
NLSY97 Appendix 2:
Employment Variable Creation

INTRODUCTION

Collection of Employment Data in Round 4. The employment sections of the NLSY97 questionnaire are somewhat complex. Before beginning analysis, researchers must understand the structure of each round's questionnaire, particularly the way in which jobs are classified as employee, freelance, or self-employment. It is important to note that this classification depends in part on the survey round and the respondent's age. In rounds 1 and 2, employee jobs were recorded in the first part of the YEMP section, administered only to respondents age 14 or older as of the interview date. The second part of the YEMP section collected information about freelance jobs of respondents age 14 and older and all jobs of respondents age 12 or 13 (the implicit assumption being that respondents younger than 14 are not likely to hold employee jobs). If the respondent was at least 16 years old and made at least \$200/week in a freelance job, the job was classified as self-employment and an extra series of questions was asked during the freelance section.

In round 3, all respondents were at least age 14 by the interview date, so the age restriction for employee jobs was no longer necessary. The structure of the section remained largely the same, with a division between employee and freelance jobs. Self-employment was classified in the same way as in the earlier rounds.

In round 4 the section was redesigned. Respondents born in 1980-82 (who were mostly age 18 and older when the round 4 field period began) were asked about employee jobs and self-employment at the same time. In addition, the minimum income requirement from the freelance section no longer applied; jobs could be classified as self-employment regardless of earnings. However, respondents born in 1983-84 (who were mostly age 16 or 17 when the round 4 field period began) continued to describe employee and freelance jobs separately. Data on self-employment jobs were still collected in the freelance section, and freelance jobs still had to meet the income criteria to qualify as self-employment.

The redesign of the employment section has important implications for created employment variables. In rounds 1-3, all of the created employment variables were based only on employee-type jobs. So, for example, the variable "Weeks Worked during Calendar Year 1999" counted only the weeks worked by a respondent at a regular employee-type job. If the respondent also reported self-employment in a lawn care business, the weeks spent working at that job were not counted in the created variable.

In round 4, when respondents born in 1980-82 reported both employee-type and self-employed jobs in the same series of variables, this approach was reconsidered. These older respondents now have three versions of most created variables. The first version, identified by the suffix "ET" in the question name, includes only employee-type jobs. The second version, the "SE" variables, includes only self-employed jobs reported by respondents born in 1980-82 in the regular employment section **during round 4**. These variables do **not** include freelance jobs or self-employment reported by younger respondents in the freelance jobs section in round 4, and they do **not** include freelance jobs or self-employment reported in previous rounds by any respondent regardless of age. Finally, the variables for all jobs include both employee-type jobs and self-employment reported during round 4 for respondents born in 1980-82 but only include employee-type jobs for respondents born in 1983-84. These last variables are identified with the suffix "ALL" in the question name.

Respondents' ages varied widely in round 4, when self-employed jobs were first recorded as part of the regular employment section, and some respondents reported employment over several years if they missed a round of interviewing. To simplify the creation of the employment variables, survey staff included only the self-employment job information starting on January 1 of the year the respondent turned 18. For example, consider a respondent who was 20 years old on his round 4 interview date in April 2001 and had not been interviewed since round 1. He reports self-employment in a computer repair business beginning on his 17th birthday in March 1998 and continuous employment at a fast-food restaurant since his round 1 interview in 1997. The round 4 created employment variables would include information about the employee-type fast food job dating all the way back to 1997. However, the computer repair business would not be considered

until January 1, 1999 (the first day of the year he turned 18). In other words, the variable “Weeks Worked in Calendar Year 1998” would count only the fast food job and the variable “Weeks Worked Any Job in Calendar Year 1999” would count both the fast food job and the repair business. Similarly, the new variable “Weeks R Was Self-Employed Year 1998” would have a value of -4, or valid skip (because the respondent was not yet 18), but the variable “Weeks R Was Self-Employed Year 1999” would report the weeks the respondent worked at the computer repair business in 1999. This approach will permit users to compare the employment variables across respondents in different rounds with confidence that the job types included are the same for all respondents of a given age.

“Backreporters.” Occasionally respondents report a job in the current interview that started before the date of their last interview and should have been reported at that time. Appendix 6 in this document contains a more complete description of the implications of these reports for the created employment event history variables. There are also a number of created employment variables detailed in this appendix. Nearly all of these variables use the information provided about employment previous to the date of last interview. The only exception is the set of CV_WKSWK_DLI variables, which reports the weeks worked since the previous interview date. The current round’s variables would not include the backreported information in any case; the previous round’s variables are **not** re-created to incorporate this new information.

For example, assume that Jane was interviewed in round 3 on April 15, 2000, and in round 4 on April 15, 2001. In the round 4 interview, she reports for the first time a job that started on April 1, 2000. The 2 weeks worked at that job before April 15, 2000, would **not** be reflected in any round’s CV_WKSWK_DLI variables. However, those weeks **would** be counted in other variables. For example, CV_WKSWK_YR.00 would count all the weeks worked at any job in 2000, regardless of whether those weeks were reported in the round 3 or round 4 interview.

Creation of Employment Variables. A number of the created employment variables use the same program as input. The program in this section is referred to throughout the employment variables. For example, to create the “Weeks Worked at Employee Job #x during 19xx” variables, survey staff first run the program below titled “emp_begin.sas” and then run the program included in the weeks worked section of this appendix.

EMP_BEGIN.SAS

This program calculates total weeks worked at each job for each respondent. It converts start and stop dates for jobs and within-job gaps to continuous week numbers, subtracts within-job gaps, and finally counts the total weeks worked. The variables listed here are those employed in any program that uses emp_begin.sas as input; they may not be used in this initial program.

Name in Program	Question Name on CD	Name in Program	Question Name on CD
Round 1			
r1int_d, _m, _y birthdy, birthmo, birthyr id	YINF-900_D, _M, _Y KEY!BDATE_D, _M, _Y PUBID	BGDY1_1, BGMO1_1, BGYR1_1 BGDY2_1, BGMO2_1, BGYR2_1 BGDY3_1, BGMO3_1, BGYR3_1 BGDY4_1, BGMO4_1, BGYR4_1 BGDY5_1, BGMO5_1, BGYR5_1 BGDY6_1, BGMO6_1, BGYR6_1 BGDY1_2, BGMO1_2, BGYR1_2 BGDY1_3, BGMO1_3, BGYR1_3 BGDY1_4, BGMO1_4, BGYR1_4 BGDY1_5, BGMO1_5, BGYR1_5 BGDY1_6, BGMO1_6, BGYR1_6 BGDY1_7, BGMO1_7, BGYR1_7 BGDY1_8, BGMO1_8, BGYR1_8 BGDY1_9, BGMO1_9, BGYR1_9 BGDY1_10, BGMO1_10, BGYR1_10 BGDY2_2, BGMO2_2, BGYR2_2 BGDY2_3, BGMO2_3, BGYR2_3	YEMP-102200.01.01-D, ~M, ~Y YEMP-102200.02.01-D, ~M, ~Y YEMP-102200.03.01-D, ~M, ~Y YEMP-102200.04.01-D, ~M, ~Y YEMP-102200.05.01-D, ~M, ~Y YEMP-102200.06.01-D, ~M, ~Y YEMP-102400.01.02-D, ~M, ~Y YEMP-102400.01.03-D, ~M, ~Y YEMP-102400.01.04-D, ~M, ~Y YEMP-102400.01.05-D, ~M, ~Y YEMP-102400.01.06-D, ~M, ~Y YEMP-102400.01.07-D, ~M, ~Y YEMP-102400.01.08-D, ~M, ~Y YEMP-102400.01.09-D, ~M, ~Y YEMP-102400.01.10-D, ~M, ~Y YEMP-102400.02.02-D, ~M, ~Y YEMP-102400.02.03-D, ~M, ~Y
Round 2	YINTDATE~D, ~M, ~Y		
r2int_d, _m, _y			
Round 3	YINTDATE~D, ~M, ~Y		
r3int_d, _m, _y			
Round 4	YINTDATE~D, ~M, ~Y		
r4int_d, _m, _y			
e200	YEMP-200A		
e303_1-4	YEMP-303.01-04		
e239011-e239019	YEMP-23901.01-09		
e245011-e245018	YEMP-24501.01-08		

Appendix 2: Employment Variable Creation

e344021-e344027	YEMP-34402.01-07	BGDY2_4, BGMO2_4, BGYR2_4	YEMP-102400.02.04~D, ~M, ~Y
e344031-e344033	YEMP-34403.01-03	BGDY2_5, BGMO2_5, BGYR2_5	YEMP-102400.02.05~D, ~M, ~Y
e344281-e344287	YEMP-34428.01-07	BGDY3_2, BGMO3_2, BGYR3_2	YEMP-102400.03.02~D, ~M, ~Y
e226101-e226103	YEMP-22610.01.02-03.02	BGDY3_3, BGMO3_3, BGYR3_3	YEMP-102400.03.03~D, ~M, ~Y
e226111-e226113	YEMP-22611.01.03-03.03	BGDY4_2, BGMO4_2, BGYR4_2	YEMP-102400.04.02~D, ~M, ~Y
e379021-e379029	YEMP-37902.01-09	BGDY4_3, BGMO4_3, BGYR4_3	YEMP-102400.04.03~D, ~M, ~Y
e379041-e379047	YEMP-37904.01-07	EGDY1_1, EGMO1_1, EGYR1_1	YEMP-102700.01.01~D, ~M, ~Y
e380001-e380009	YEMP-38000.01-09	EGDY1_2, EGMO1_2, EGYR1_2	YEMP-102700.01.02~D, ~M, ~Y
e38000F1-e38000F7	YEMP-38000F.01-07	EGDY1_3, EGMO1_3, EGYR1_3	YEMP-102700.01.03~D, ~M, ~Y
e381031-e381033	YEMP-38103.01-03	EGDY1_4, EGMO1_4, EGYR1_4	YEMP-102700.01.04~D, ~M, ~Y
e381051-e381055	YEMP-38105.01-05	EGDY1_5, EGMO1_5, EGYR1_5	YEMP-102700.01.05~D, ~M, ~Y
e599011-e599018	YEMP-59901.01-08	EGDY1_6, EGMO1_6, EGYR1_6	YEMP-102700.01.06~D, ~M, ~Y
e880001-e880009	YEMP-88000.01-09	EGDY1_7, EGMO1_7, EGYR1_7	YEMP-102700.01.07~D, ~M, ~Y
e885011-e885017	YEMP-88501.01-07	EGDY1_8, EGMO1_8, EGYR1_8	YEMP-102700.01.08~D, ~M, ~Y
e984021-e984027	YEMP-98402.01-07	EGDY1_9, EGMO1_9, EGYR1_9	YEMP-102700.01.09~D, ~M, ~Y
e984031-e984033	YEMP-98403.01-03	EGDY1_10, EGMO1_10, EGYR1_10	YEMP-102700.01.10~D, ~M, ~Y
e984291-e984293	YEMP-98429.01-03	EGDY2_1, EGMO2_1, EGYR2_1	YEMP-102700.02.01~D, ~M, ~Y
e1002311-e1002314	YEMP-100231.01.02-04.02	EGDY2_2, EGMO2_2, EGYR2_2	YEMP-102700.02.02~D, ~M, ~Y
e1002321-e1002323	YEMP-100232.01.03-03.03	EGDY2_3, EGMO2_3, EGYR2_3	YEMP-102700.02.03~D, ~M, ~Y
GAP1-GAP9	YEMP-101500.01-09	EGDY2_4, EGMO2_4, EGYR2_4	YEMP-102700.02.04~D, ~M, ~Y
NUMGAP1-NUMGAP6	YEMP-103200.01-06	EGDY2_5, EGMO2_5, EGYR2_5	YEMP-102700.02.05~D, ~M, ~Y
milflag1-milflag9	YEMP_MILFLAG.01-09	EGDY3_1, EGMO3_1, EGYR3_1	YEMP-102700.03.01~D, ~M, ~Y
self1-self9	YEMP_SELFEMP.01-09	EGDY3_2, EGMO3_2, EGYR3_2	YEMP-102700.03.02~D, ~M, ~Y
stardy1-stardy9	YEMP_STARTDATE.01~D..09~D	EGDY3_3, EGMO3_3, EGYR3_3	YEMP-102700.03.03~D, ~M, ~Y
starmo1-starmo9	YEMP_STARTDATE.01~M..09~M	EGDY4_1, EGMO4_1, EGYR4_1	YEMP-102700.04.01~D, ~M, ~Y
staryr1-staryr9	YEMP_STARTDATE.01~Y..09~Y	EGDY4_2, EGMO4_2, EGYR4_2	YEMP-102700.04.02~D, ~M, ~Y
stopdy1-stopdy9	YEMP_STOPDATE.01~D..09~D	EGDY4_3, EGMO4_3, EGYR4_3	YEMP-102700.04.03~D, ~M, ~Y
stopmo1-stopmo9	YEMP_STOPDATE.01~M..09~M	EGDY5_1, EGMO5_1, EGYR5_1	YEMP-102700.05.01~D, ~M, ~Y
stopyr1-stopyr9	YEMP_STOPDATE.01~Y..09~Y	EGDY6_1, EGMO6_1, EGYR6_1	YEMP-102700.06.01~D, ~M, ~Y
uid1-uid9	YEMP_UID.01-09		

***** SECTION 1: Convert birth week, week turned age 14, week turned age 20, *****
***** and week of interview date into continuous weeks *****

```

/** Calulate Age 14 year **/ AGE14YR=birthyr+14;
** Convert Age 14 Birthdate to week number *** Convert age14 month and day to total days (bbdays);
if birthmo>0 and birthdy>0 then do;
  if birthmo=1 then bbdays=birthdy;
  if birthmo=3 then bbdays=birthdy+59;
  if birthmo=5 then bbdays=birthdy+120;
  if birthmo=7 then bbdays=birthdy+181;
  if birthmo=9 then bbdays=birthdy+243;
  if birthmo=11 then bbdays=birthdy+304;
end;

***Account for leap years;
if age14yr=1980 or age14yr=1984 or age14yr=1988 or age14yr=1992 or age14yr=1996 or age18yr=2000 then do;
  if birthmo>0 and birthdy>0 then do;
    if birthmo=1 then bbdays=birthdy;
    if birthmo=3 then bbdays=birthdy+60;
    if birthmo=5 then bbdays=birthdy+121;
    if birthmo=7 then bbdays=birthdy+182;
    if birthmo=9 then bbdays=birthdy+244;
    if birthmo=11 then bbdays=birthdy+305;
  end;
end;

***Convert days into week numbers;
/** Basic Formula: weekno=endweek{specific year}+ceil[(totdays+{# of days remaining in DEC})/7] **
** This applies throughout program whenever days are converted into week numbers **/

```

Appendix 2: Employment Variable Creation

```

/* Default age 14 week = 9999 */
age14wk=9999;
if age14yr>0 and bbdays>0 then do;
  if age14yr=1980 then do; age14wk=ceil((bbdays+2)/7); end;
  if age14yr=1981 then do; age14wk=52+ceil((bbdays+4)/7); end;
  if age14yr=1982 then do; age14wk=104+ceil((bbdays+5)/7); end;
  if age14yr=1983 then do; age14wk=156+ceil((bbdays+6)/7); end;
  if age14yr=1984 then do; age14wk=209+ceil((bbdays)/7); end;
  if age14yr=1985 then do; age14wk=261+ceil((bbdays+2)/7); end;
  if age14yr=1986 then do; age14wk=313+ceil((bbdays+3)/7); end;
  if age14yr=1987 then do; age14wk=365+ceil((bbdays+4)/7); end;
  if age14yr=1988 then do; age14wk=417+ceil((bbdays+5)/7); end;
  if age14yr=1989 then do; age14wk=470+ceil((bbdays)/7); end;
  if age14yr=1990 then do; age14wk=522+ceil((bbdays+1)/7); end;
  if age14yr=1991 then do; age14wk=574+ceil((bbdays+2)/7); end;
  if age14yr=1992 then do; age14wk=626+ceil((bbdays+3)/7); end;
  if age14yr=1993 then do; age14wk=678+ceil((bbdays+5)/7); end;
  if age14yr=1994 then do; age14wk=730+ceil((bbdays+6)/7); end;
  if age14yr=1995 then do; age14wk=783+ceil((bbdays)/7); end;
  if age14yr=1996 then do; age14wk=835+ceil((bbdays+1)/7); end;
  if age14yr=1997 then do; age14wk=887+ceil((bbdays+3)/7); end;
  if age14yr=1998 then do; age14wk=939+ceil((bbdays+4)/7); end;
  if age14yr=1999 then do; age14wk=991+ceil((bbdays+5)/7); end;
  if age14yr=2000 then do; age14wk=1043+ceil((bbdays+6)/7); end;
end;

/** Calulate Age 18 year **/ AGE18YR=birthyr+18;
** Convert Age 18 Birthdate to week number ** ***Convert age18 month and day to total days (bbdays);
if birthmo>0 and birthdy>0 then do;
  if birthmo=1 then bbdays=birthdy;
  if birthmo=3 then bbdays=birthdy+59;
  if birthmo=5 then bbdays=birthdy+120;
  if birthmo=7 then bbdays=birthdy+181;
  if birthmo=9 then bbdays=birthdy+243;
  if birthmo=11 then bbdays=birthdy+304;
  if birthmo=2 then bbdays=birthdy+31;
  if birthmo=4 then bbdays=birthdy+90;
  if birthmo=6 then bbdays=birthdy+151;
  if birthmo=8 then bbdays=birthdy+212;
  if birthmo=10 then bbdays=birthdy+273;
  if birthmo=12 then bbdays=birthdy+334;
end;

***Account for leap years;
if age18yr=1980 or age18yr=1984 or age18yr=1988 or age18yr=1992 or age18yr=1996 or age18yr=2000 then do;
  if birthmo>0 and birthdy>0 then do;
    if birthmo=1 then bbdays=birthdy;
    if birthmo=3 then bbdays=birthdy+60;
    if birthmo=5 then bbdays=birthdy+121;
    if birthmo=7 then bbdays=birthdy+182;
    if birthmo=9 then bbdays=birthdy+244;
    if birthmo=11 then bbdays=birthdy+305;
    if birthmo=2 then bbdays=birthdy+31;
    if birthmo=4 then bbdays=birthdy+91;
    if birthmo=6 then bbdays=birthdy+152;
    if birthmo=8 then bbdays=birthdy+213;
    if birthmo=10 then bbdays=birthdy+274;
    if birthmo=12 then bbdays=birthdy+335;
  end;
end;

***Convert days into week numbers; /* Default age 14 week = 9999 */
age18wk=9999;
if age18yr>0 and bbdays>0 then do;
  if age18yr=1980 then do; age18wk=ceil((bbdays+2)/7); ag18jan=0; end;
  if age18yr=1981 then do; age18wk=52+ceil((bbdays+4)/7); ag18jan=52; end;
  if age18yr=1982 then do; age18wk=104+ceil((bbdays+5)/7); ag18jan=104; end;
  if age18yr=1983 then do; age18wk=156+ceil((bbdays+6)/7); ag18jan=156; end;
  if age18yr=1984 then do; age18wk=209+ceil((bbdays)/7); ag18jan=209; end;

```

```

if age18yr=1985 then do; age18wk=261+ceil((bbdays+2)/7); ag18jan=261; end;
if age18yr=1986 then do; age18wk=313+ceil((bbdays+3)/7); ag18jan=313; end;
if age18yr=1987 then do; age18wk=365+ceil((bbdays+4)/7); ag18jan=365; end;
if age18yr=1988 then do; age18wk=417+ceil((bbdays+5)/7); ag18jan=417; end;
if age18yr=1989 then do; age18wk=470+ceil((bbdays)/7); ag18jan=470; end;
if age18yr=1990 then do; age18wk=522+ceil((bbdays+1)/7); ag18jan=522; end;
if age18yr=1991 then do; age18wk=574+ceil((bbdays+2)/7); ag18jan=574; end;
if age18yr=1992 then do; age18wk=626+ceil((bbdays+3)/7); ag18jan=626; end;
if age18yr=1993 then do; age18wk=678+ceil((bbdays+5)/7); ag18jan=678; end;
if age18yr=1994 then do; age18wk=730+ceil((bbdays+6)/7); ag18jan=730; end;
if age18yr=1995 then do; age18wk=783+ceil((bbdays)/7); ag18jan=784; end;
if age18yr=1996 then do; age18wk=835+ceil((bbdays+1)/7); ag18jan=836; end;
if age18yr=1997 then do; age18wk=887+ceil((bbdays+3)/7); ag18jan=888; end;
if age18yr=1998 then do; age18wk=939+ceil((bbdays+4)/7); ag18jan=940; end;
if age18yr=1999 then do; age18wk=991+ceil((bbdays+5)/7); ag18jan=992; end;
if age18yr=2000 then do; age18wk=1043+ceil((bbdays+6)/7); ag18jan=1044; end;
if age18yr=2001 then do; age18wk=1096+ceil((bbdays+1)/7); ag18jan=1097; end;
if age18yr=2002 then do; age18wk=1148+ceil((bbdays+2)/7); ag18jan=1149; end;
end;

/** Calulate Age 20 week **/ AGE20YR=birthyr+20;
** Convert Age 20 Birthdate to week number ** ***Convert age20 month and day to total days (bbdays);
if birthmo>0 and birthdy>0 then do;
  if birthmo=1 then bbdays=birthdy;
  if birthmo=3 then bbdays=birthdy+59;
  if birthmo=5 then bbdays=birthdy+120;
  if birthmo=7 then bbdays=birthdy+181;
  if birthmo=9 then bbdays=birthdy+243;
  if birthmo=11 then bbdays=birthdy+304;
  if birthmo=2 then bbdays=birthdy+31;
  if birthmo=4 then bbdays=birthdy+90;
  if birthmo=6 then bbdays=birthdy+151;
  if birthmo=8 then bbdays=birthdy+212;
  if birthmo=10 then bbdays=birthdy+273;
  if birthmo=12 then bbdays=birthdy+334;
end;

***Account for leap years;
if age18yr=1980 or age18yr=1984 or age18yr=1988 or age18yr=1992 or age18yr=1996 or age18yr=2000 then do;
  if birthmo>0 and birthdy>0 then do;
    if birthmo=1 then bbdays=birthdy;
    if birthmo=3 then bbdays=birthdy+60;
    if birthmo=5 then bbdays=birthdy+121;
    if birthmo=7 then bbdays=birthdy+182;
    if birthmo=9 then bbdays=birthdy+244;
    if birthmo=11 then bbdays=birthdy+305;
    if birthmo=2 then bbdays=birthdy+31;
    if birthmo=4 then bbdays=birthdy+91;
    if birthmo=6 then bbdays=birthdy+152;
    if birthmo=8 then bbdays=birthdy+213;
    if birthmo=10 then bbdays=birthdy+274;
    if birthmo=12 then bbdays=birthdy+335;
  end;
end;

***Convert days into week numbers; /* Default age 20 week = 9999 */
age20wk=9999;
if age20yr>0 and bbdays>0 then do;
  if age20yr=1980 then do; age20wk=ceil((bbdays+2)/7); end;
  if age20yr=1981 then do; age20wk=52+ceil((bbdays+4)/7); end;
  if age20yr=1982 then do; age20wk=104+ceil((bbdays+5)/7); end;
  if age20yr=1983 then do; age20wk=156+ceil((bbdays+6)/7); end;
  if age20yr=1984 then do; age20wk=209+ceil((bbdays)/7); end;
  if age20yr=1985 then do; age20wk=261+ceil((bbdays+2)/7); end;
  if age20yr=1986 then do; age20wk=313+ceil((bbdays+3)/7); end;
  if age20yr=1987 then do; age20wk=365+ceil((bbdays+4)/7); end;
  if age20yr=1988 then do; age20wk=417+ceil((bbdays+5)/7); end;
  if age20yr=1989 then do; age20wk=470+ceil((bbdays)/7); end;
  if age20yr=1990 then do; age20wk=522+ceil((bbdays+1)/7); end;

```

Appendix 2: Employment Variable Creation

```
if age20yr=1991 then do; age20wk=574+ceil((bbdays+2)/7); end;
if age20yr=1992 then do; age20wk=626+ceil((bbdays+3)/7); end;
if age20yr=1993 then do; age20wk=678+ceil((bbdays+5)/7); end;
if age20yr=1994 then do; age20wk=730+ceil((bbdays+6)/7); end;
if age20yr=1995 then do; age20wk=783+ceil((bbdays)/7); end;
if age20yr=1996 then do; age20wk=835+ceil((bbdays+1)/7); end;
if age20yr=1997 then do; age20wk=887+ceil((bbdays+3)/7); end;
if age20yr=1998 then do; age20wk=939+ceil((bbdays+4)/7); end;
if age20yr=1999 then do; age20wk=991+ceil((bbdays+5)/7); end;
if age20yr=2000 then do; age20wk=1043+ceil((bbdays+6)/7); end;
if age20yr=2001 then do; age20wk=1096+ceil((bbdays+1)/7); end;
if age20yr=2002 then do; age20wk=1148+ceil((bbdays+2)/7); end;
if age20yr=2003 then do; age20wk=1200+ceil((bbdays+2)/7); end;
if age20yr=2004 then do; age20wk=1252+ceil((bbdays+2)/7); end;
end;

/** Convert Birthdate to week number */ /* Default birthdate week=0 if birthdate < 12/30/79 */
birthwk=0;
if birthyr>0 and bbdays>0 then do;
  if birthyr=1980 then do; birthwk=ceil((bbdays+2)/7); end;
  if birthyr=1981 then do; birthwk=52+ceil((bbdays+4)/7); end;
  if birthyr=1982 then do; birthwk=104+ceil((bbdays+5)/7); end;
  if birthyr=1983 then do; birthwk=156+ceil((bbdays+6)/7); end;
  if birthyr=1984 then do; birthwk=209+ceil((bbdays)/7); end;
  if birthyr=1985 then do; birthwk=261+ceil((bbdays+2)/7); end;
  if birthyr=1986 then do; birthwk=313+ceil((bbdays+3)/7); end;
  if birthyr=1987 then do; birthwk=365+ceil((bbdays+4)/7); end;
  if birthyr=1988 then do; birthwk=417+ceil((bbdays+5)/7); end;
  if birthyr=1989 then do; birthwk=470+ceil((bbdays)/7); end;
  if birthyr=1990 then do; birthwk=522+ceil((bbdays+1)/7); end;
  if birthyr=1991 then do; birthwk=574+ceil((bbdays+2)/7); end;
  if birthyr=1992 then do; birthwk=626+ceil((bbdays+3)/7); end;
  if birthyr=1993 then do; birthwk=678+ceil((bbdays+5)/7); end;
  if birthyr=1994 then do; birthwk=730+ceil((bbdays+6)/7); end;
  if birthyr=1995 then do; birthwk=783+ceil((bbdays)/7); end;
  if birthyr=1996 then do; birthwk=835+ceil((bbdays+1)/7); end;
  if birthyr=1997 then do; birthwk=887+ceil((bbdays+3)/7); end;
  if birthyr=1998 then do; birthwk=939+ceil((bbdays+4)/7); end;
  if birthyr=1999 then do; birthwk=991+ceil((bbdays+5)/7); end;
  if birthyr=2000 then do; birthyr=1043+ceil((bbdays+6)/7); end;
end;

***** Section 2: Convert Interview Dates to week number *****/
***Convert Round 4 interview month and day to total days (intdays);
if r4int_m>0 and r4int_d>0 then do;
  if r4int_m=1 then intdays=r4int_d;                                if r4int_m=2 then intdays=r4int_d+31;
  if r4int_m=3 then intdays=r4int_d+59;                            if r4int_m=4 then intdays=r4int_d+90;
  if r4int_m=5 then intdays=r4int_d+120;                           if r4int_m=6 then intdays=r4int_d+151;
  if r4int_m=7 then intdays=r4int_d+181;                           if r4int_m=8 then intdays=r4int_d+212;
  if r4int_m=9 then intdays=r4int_d+243;                           if r4int_m=10 then intdays=r4int_d+273;
  if r4int_m=11 then intdays=r4int_d+304;                           if r4int_m=12 then intdays=r4int_d+334;
end;

***Account for leap year 2000;
if r4int_m>0 and r4int_d>0 and r4int_y=2000 then do;
  if r4int_m=1 then intdays=r4int_d;                                if r4int_m=2 then intdays=r4int_d+31;
  if r4int_m=3 then intdays=r4int_d+60;                            if r4int_m=4 then intdays=r4int_d+91;
```

Appendix 2: Employment Variable Creation

```

if r4int_m=5 then intdays=r4int_d+121;
if r4int_m=7 then intdays=r4int_d+182;
if r4int_m=9 then intdays=r4int_d+244;
if r4int_m=11 then intdays=r4int_d+305;
end;

***Convert days into week numbers; /* Default interview week = 9999 */
intwk=9999;
if r4int_y>0 and intdays>0 then do;
  if r4int_y=2000 then do; intwk=1043+ceil((intdays+6)/7); end;
  if r4int_y=2001 then do; intwk=1096+ceil((intdays+1)/7); end;
end;

** Convert Round 3 Interview Date to week number **; ***Convert interview month and day to total days (intdays);
if r3int_m>0 and r3int_d>0 then do;
  if r3int_m=1 then intdays=r3int_d;
  if r3int_m=3 then intdays=r3int_d+59;
  if r3int_m=5 then intdays=r3int_d+120;
  if r3int_m=7 then intdays=r3int_d+181;
  if r3int_m=9 then intdays=r3int_d+243;
  if r3int_m=11 then intdays=r3int_d+304;
end;

if r3int_m=2 then intdays=r3int_d+31;
if r3int_m=4 then intdays=r3int_d+90;
if r3int_m=6 then intdays=r3int_d+151;
if r3int_m=8 then intdays=r3int_d+212;
if r3int_m=10 then intdays=r3int_d+273;
if r3int_m=12 then intdays=r3int_d+334;

***Account for leap year 2000;
if r3int_m>0 and r3int_d>0 and r3int_y=2000 then do;
  if r3int_m=1 then intdays=r3int_d;
  if r3int_m=3 then intdays=r3int_d+60;
  if r3int_m=5 then intdays=r3int_d+121;
  if r3int_m=7 then intdays=r3int_d+182;
  if r3int_m=9 then intdays=r3int_d+244;
  if r3int_m=11 then intdays=r3int_d+305;
end;

if r3int_m=2 then intdays=r3int_d+31;
if r3int_m=4 then intdays=r3int_d+91;
if r3int_m=6 then intdays=r3int_d+152;
if r3int_m=8 then intdays=r3int_d+213;
if r3int_m=10 then intdays=r3int_d+274;
if r3int_m=12 then intdays=r3int_d+335;

***Convert days into week numbers; /* Default interview week = 9999 */
r3int=9999;
if r3int_y>0 and intdays>0 then do;
  if r3int_y=1999 then do; r3int=991+ceil((intdays+5)/7); end;
  if r3int_y=2000 then do; r3int=1043+ceil((intdays+6)/7); end;
end;

** Convert Round 2 Date of Interview to week number ** ***Convert interview month and day to total days (rd2days);
if r2int_m>0 and r2int_d>0 then do;
  if r2int_m=1 then rd2days=r2int_d;
  if r2int_m=3 then rd2days=r2int_d+59;
  if r2int_m=5 then rd2days=r2int_d+120;
  if r2int_m=7 then rd2days=r2int_d+181;
  if r2int_m=9 then rd2days=r2int_d+243;
  if r2int_m=11 then rd2days=r2int_d+304;
end;

if r2int_m=2 then rd2days=r2int_d+31;
if r2int_m=4 then rd2days=r2int_d+90;
if r2int_m=6 then rd2days=r2int_d+151;
if r2int_m=8 then rd2days=r2int_d+212;
if r2int_m=10 then rd2days=r2int_d+273;
if r2int_m=12 then rd2days=r2int_d+334;

***Convert days into week numbers; /* Default interview week = 9999 */
r2int=9999;
if r2int_y>0 and rd2days>0 then do;
  if r2int_y=1998 then do; r2int=939+ceil((rd2days+4)/7); end;
  if r2int_y=1999 then do; r2int=991+ceil((rd2days+5)/7); end;
end;

```

```

** Convert Date of Round 1 Interview to week number **Convert interview month and day to total days (rd2days);
if r1int_m>0 and r1int_d>0 then do;
  if r1int_m=1 then rd1days=r1int_d;
  if r1int_m=3 then rd1days=r1int_d+59;
  if r1int_m=5 then rd1days=r1int_d+120;
  if r1int_m=7 then rd1days=r1int_d+181;
  if r1int_m=9 then rd1days=r1int_d+243;
  if r1int_m=11 then rd1days=r1int_d+304;
  if r1int_m=2 then rd1days=r1int_d+31;
  if r1int_m=4 then rd1days=r1int_d+90;
  if r1int_m=6 then rd1days=r1int_d+151;
  if r1int_m=8 then rd1days=r1int_d+212;
  if r1int_m=10 then rd1days=r1int_d+273;
  if r1int_m=12 then rd1days=r1int_d+334;
end;

***Convert days into week numbers; /* Default interview week = 9999 */
r1int=9999;
if r1int_y>0 and rd1days>0 then do;
  if r1int_y=1997 then do; r1int=887+ceil((rd1days+3)/7); end;
  if r1int_y=1998 then do; r1int=939+ceil((rd1days+4)/7); end;
end;

/* Hand edits carried over from Round 2 */
if (id=1000291 or id=1562991 or id=1590571) then do; r1int_y=1997; end;
if id in (1057531,1058331) then do; r2int=990; r2int_y=1998; end;
if id=1745011 then do; r1int=919; r1int_y=1997; end;

/* Done to create a seamless flow between rounds */
intwk=intwk-1;

/* Round 2 non-interview case */ if r2int_d=-5 then r2int=-5;
/* Round 3 non-interview case */ if r3int_d=-5 then r3int=-5;
/* Round 4 non-interview case */ if e200=-5 then intwk=-5;

***** Section 3: Convert Start/Stop dates into NLSY97 week numbers *****
/* This section reads in raw start and stop dates for each job (max=9) and converts them into NLSY97
week numbers. Some start/stop DAYS and MONTHS have been imputed if missing. */
/*ostartm represents the "old" start dates, used when start dates are updated to interview dates. */
array ostarm (i) ostarmo1-ostarmo9;           array ostard (i) ostardy1-ostardy9;
array ostary (i) ostaryr1-ostaryr9;           array ostopm (i) ostopmo1-ostopmo9;
array ostopd (i) ostopdy1-ostopdy9;           array ostopy (i) ostopyr1-ostopyr9;
array startm (i) starmo1-starmo9;             array startd (i) stardy1-stardy9;
array starty (i) staryr1-star yr9;            array stopm (i) stopmo1-stopmo9;
array stopd (i) stopdy1-stopdy9;                array stopy (i) stopyr1-stopyr9;
array UID (i) UID1-UID9;
array sttdays (i) sttday1-sttday9; /* total days in that year from startdate (to Jan 1) */
array stpdays (i) stpday1-stpday9; /* total days in that year from stopdate (to Jan 1) */
array startwk (i) starw1-starw9;               array stopwk (i) stopw1-stopw9;
array srflag (i) srflg1-srflg10;              array spflag (i) spflg1-spflg10;
array uflag (i) uflag1-uflag9; /* uflag=1 when job startdate is updated */
array smofl (i) smofl1-smofl9; /* dummy equals 1 when a start month is imputed */
array emofl (i) emofl1-emofl9; /* dummy equals 1 when a stop month is imputed */
array self (i) self1-self9;

/* Establish dli interview round */
if r3int_y>0 then do; dli_y=r3int_y; dli_m=r3int_m; dli_d=r3int_d; dliwk=r3int; r3dli=1; end;
if r3int_y=-5 and r2int_y>0 then do; dli_y=r2int_y; dli_m=r2int_m; dli_d=r2int_d; dliwk=r2int; r2dli=1; end;
if r3int_y=-5 and r2int_y=-5 and r1int_y>0 then do; dli_y=r1int_y; dli_m=r1int_m; dli_d=r1int_d; dliwk=r1int;
  r1dli=1; end;

/* Initialize smofl and emofl */ do i=1 to 9; smofl=0; emofl=0; end;

```

```

/* Define old start and stop dates */
do i=1 to 9; ostartm=startm; ostartd=startd; ostarty=starty; ostopm=stopm; ostopd=stopd; ostopy=stopy; end;

***Fill-in start/stop day for those missing;
/* flag1 = impute start day (valid month)      flag2 = impute start month (valid day)
   flag3 = impute start day and month        flag4 = impute stop day (valid month)
   flag5 = impute stop month (valid day)       flag6 = impute stop day and month      */
do i=1 to 9; flag1=-4; flag2=-4; flag3=-4; flag4=-4; flag5=-4; flag6=-4; end;

/* Impute missing start days to 1, missing stop days to 28, missing start months to 1 (Jan.) and missing stop months to
   12 (Dec.). */ /* Reset flag(1-3) to zero when the start year is valid */
do over starty;
if starty>0 then do;
  flag1=0; flag2=0; flag3=0;
  if startm>0 and startd<=0 then do; startd=1; flag1=1; srflag=1; end;
  if startm<=0 and startd>0 then do; startm=1; flag2=1; srflag=1; smofl=1; end;
  if startm<=0 and startd<=0 then do; startm=1; startd=1; flag3=1; srflag=1; smofl=1; end;
end;
end;

do over stopy;
if stopy>0 then do;
  flag4=0; flag5=0; flag6=0;
  if stopm>0 and stopd<=0 then do; stopd=28; flag4=1; spflag=1; end;
  if stopm<=0 and stopd>0 then do; stopm=12; flag5=1; spflag=1; emofl=1; end;
  if stopm<=0 and stopd<=0 then do; stopm=12; stopd=28; flag6=1; spflag=1; emofl=1; end;
/* The following lines prevent imputed values for stop months and days to exceed the Round4 interview date. */
  if stopy=r4int_y and stopm=r4int_m and stopd>r4int_d and spflag=1 then do; stopd=r4int_d; end;
  if stopy=r4int_y and stopm>r4int_m then do; stopm=r4int_m; stopd=r4int_d; end;
end;
end;

/* If a respondent backreports a job that begins before the dli but ends after, then the start date is updated to the dli. If a
   respondent backreports a job that begins and ends before the dli, that job is eliminated. A list of the deleted
   jobs is below. Deletion occurs after the start/stop dates are turned into week numbers later in the program.
   The id's of the backreporters are listed below. */
do i=1 to 9;
if 0<starty<dli_y and 0<stopy<dli_y then brep=1;
if starty<dli_y and stopy=dli_y and 0<stopm<dli_m then brep=1;
if starty=dli_y and stopy=dli_y and 0<startm<dli_m and 0<stopm<dli_m then brep=1;
if starty=dli_y and stopy=dli_y and 0<startm<dli_m and stopm=dli_m and 0<stopd<dli_d then brep=1;
if starty=dli_y and stopy=dli_y and startm=dli_m and stopm=dli_m and 0<startd<dli_d and 0<stopd<dli_d then
  brep=1;
end;

array backr (i) backr1-backr9;
do i=1 to 9; backr=0; end;
if id=1001101 and uid2=200001 then backr2=1;           if id=1036631 and uid2=200001 then backr2=1;
if id=1094982 and uid3=200001 then backr3=1;           if id=1206521 and uid1=200001 then backr1=1;
if id=1243411 and uid4=200004 then backr4=1;           if id=1243411 and uid5=200005 then backr5=1;
if id=1283021 and uid4=200001 then backr4=1;           if id=1370661 and uid3=200001 then backr3=1;
if id=1438351 and uid2=200002 then backr2=1;           if id=1445111 and uid2=200002 then backr2=1;
if id=1479601 and uid3=200002 then backr3=1;           if id=1549682 and uid4=200003 then backr4=1;
if id=1558701 and uid2=200002 then backr2=1;           if id=1674502 and uid2=200002 then backr2=1;
if id=1674502 and uid3=200001 then backr3=1;           if id=1718641 and uid3=200001 then backr3=1;
if id=1882171 and uid1=200001 then backr1=1;           if id=1927192 and uid1=200001 then backr1=1;

```

Appendix 2: Employment Variable Creation

```
if id=1955071 and uid2=200001 then backr2=1;

/* We can erase one backreporting job for id 1622011, uid=9802 since we already have data on that job from Round 3.
   This leaves a total of 17 backreporting respondents and 19 backreporting jobs, all of which have 2000 UID's.
   This data will be used in the job-specific c.v.'s, but not the year specific or age 14 c.v.'s. */

/* id 1283021 is a backerporting respondent who has the job erased later in the program because it is a self-employed
   job that begins and ends before Jan.1, of the year of R's 18th birthday. */

/* id 1704581 reports a 1999 job that begins and ends before dli. */

if id=1622011 or id=1704581 then do;
  stardy1=-4; starmo1=-4; starty1=-4; stopd1=-4; stopmo1=-4; stopyr1=-4; end;

/* The following lines of code account for the cases where the respondent has a start date for a job reported in an earlier
   round that comes before the Round 3 (dli) interview date. The idea is to only count the weeks employed from
   dli to today. Then, the activity from previous rounds will be added to the Round 4 activity to get the full
   history. To achieve this, all jobs reported in earlier rounds (UID's begin with 97,98,or 1999) will have their
   startdates updated to the Ruond 3 (dli) interview date. The start week and stop week for these jobs will be
   counted the same. */

do i=1 to 9;
  uflag=0; /* Initialize uflag */
  if uid>-5 and startd>-4 and backr=0 then do;
    if starty=dli_y and startm=dli_m and startd<dli_d and starty>0 then do; startd=dli_d; uflag=1; end;
    if starty=dli_y and startm<=dli_m and starty>0 then do; startm=dli_m; startd=dli_d; uflag=1; end;
    if starty<dli_y and starty>0 then do; starty=dli_y; startm=dli_m; startd=dli_d; uflag=1; end;
  end;
end;

***Convert START month and day to total days;
do over startm;
  if startm>0 and startd>0 then do;
    if startm=1 then sttdays=startd;
    if startm=3 then sttdays=startd+59;
    if startm=5 then sttdays=startd+120;
    if startm=7 then sttdays=startd+181;
    if startm=9 then sttdays=startd+243;
    if startm=11 then sttdays=startd+304;
  end;
end;

***Account for leap years;
do over starty;
  if starty=1980 or starty=1984 or starty=1988 or starty=1992 or starty=1996 or starty=2000 then do;
    if startm>0 and startd>0 then do;
      if startm=1 then sttdays=startd;
      if startm=3 then sttdays=startd+60;
      if startm=5 then sttdays=startd+121;
      if startm=7 then sttdays=startd+182;
      if startm=9 then sttdays=startd+244;
      if startm=11 then sttdays=startd+305;
    end;
  end;
end;

***Convert STOP month and day to total days;
do over stopm;
  if stopm>0 and stopd>0 then do;
    if stopm=1 then stpdays=stopd;
    if stopm=3 then stpdays=stopd+59;
  end;
  if stopm=2 then stpdays=stopd+31;
  if stopm=4 then stpdays=stopd+90;
```

```

if stopm=5 then stpdays=stopd+120;
if stopm=7 then stpdays=stopd+181;
if stopm=9 then stpdays=stopd+243;
if stopm=11 then stpdays=stopd+304;
end;
end;

***Account for leap years;
do over stopy;
if stopy=1980 or stopy=1984 or stopy=1988 or stopy=1992 or stopy=1996 or stopy=2000 then do;
if stopm>0 and stopd>0 then do;
if stopm=1 then stpdays=stopd;
if stopm=3 then stpdays=stopd+60;
if stopm=5 then stpdays=stopd+121;
if stopm=7 then stpdays=stopd+182;
if stopm=9 then stpdays=stopd+244;
if stopm=11 then stpdays=stopd+305;
end;
end;
end;

***Convert days into week numbers;
/* Create year flag variable */
array byear (i) byear1-byear9;
array eyear (i) eyear1-eyear9;
do over starty;
if starty>0 and sttdays>0 then do;
if starty=1980 then do; startwk=ceil((sttdays+2)/7); end;
if starty=1981 then do; startwk=52+ceil((sttdays+4)/7); end;
if starty=1982 then do; startwk=104+ceil((sttdays+5)/7); end;
if starty=1983 then do; startwk=156+ceil((sttdays+6)/7); end;
if starty=1984 then do; startwk=209+ceil((sttdays)/7); end;
if starty=1985 then do; startwk=261+ceil((sttdays+2)/7); end;
if starty=1986 then do; startwk=313+ceil((sttdays+3)/7); end;
if starty=1987 then do; startwk=365+ceil((sttdays+4)/7); end;
if starty=1988 then do; startwk=417+ceil((sttdays+5)/7); end;
if starty=1989 then do; startwk=470+ceil((sttdays)/7); end;
if starty=1990 then do; startwk=522+ceil((sttdays+1)/7); end;
if starty=1991 then do; startwk=574+ceil((sttdays+2)/7); end;
if starty=1992 then do; startwk=626+ceil((sttdays+3)/7); end;
if starty=1993 then do; startwk=678+ceil((sttdays+5)/7); end;
if starty=1994 then do; startwk=730+ceil((sttdays+6)/7); end;
if starty=1995 then do; startwk=783+ceil((sttdays)/7); end;
if starty=1996 then do; startwk=835+ceil((sttdays+1)/7); end;
if starty=1997 then do; startwk=887+ceil((sttdays+3)/7); end;
if starty=1998 then do; startwk=939+ceil((sttdays+4)/7); end;
if starty=1999 then do; startwk=991+ceil((sttdays+5)/7); end;
if starty=2000 then do; startwk=1043+ceil((sttdays+6)/7); end;
if starty=2001 then do; startwk=1096+ceil((sttdays+1)/7); end;
end;
if starty<0 and starty>-4 then do; startwk=-3; end;
if -4<ostarty<0 then byear=1;
end;

do over stopy;
if stopy>0 and stpdays>0 then do;
if stopy=1980 then do; stopwk=ceil((stpdays+2)/7); end;
if stopm=6 then stpdays=stopd+151;
if stopm=8 then stpdays=stopd+212;
if stopm=10 then stpdays=stopd+273;
if stopm=12 then stpdays=stopd+334;

```

Appendix 2: Employment Variable Creation

```
if stopy=1981 then do; stopwk=52+ceil((stpdays+4)/7); end;
if stopy=1982 then do; stopwk=104+ceil((stpdays+5)/7); end;
if stopy=1983 then do; stopwk=156+ceil((stpdays+6)/7); end;
if stopy=1984 then do; stopwk=209+ceil((stpdays)/7); end;
if stopy=1985 then do; stopwk=261+ceil((stpdays+2)/7); end;
if stopy=1986 then do; stopwk=313+ceil((stpdays+3)/7); end;
if stopy=1987 then do; stopwk=365+ceil((stpdays+4)/7); end;
if stopy=1988 then do; stopwk=417+ceil((stpdays+5)/7); end;
if stopy=1989 then do; stopwk=470+ceil((stpdays)/7); end;
if stopy=1990 then do; stopwk=522+ceil((stpdays+1)/7); end;
if stopy=1991 then do; stopwk=574+ceil((stpdays+2)/7); end;
if stopy=1992 then do; stopwk=626+ceil((stpdays+3)/7); end;
if stopy=1993 then do; stopwk=678+ceil((stpdays+5)/7); end;
if stopy=1994 then do; stopwk=730+ceil((stpdays+6)/7); end;
if stopy=1995 then do; stopwk=783+ceil((stpdays)/7); end;
if stopy=1996 then do; stopwk=835+ceil((stpdays+1)/7); end;
if stopy=1997 then do; stopwk=887+ceil((stpdays+3)/7); end;
if stopy=1998 then do; stopwk=939+ceil((stpdays+4)/7); end;
if stopy=1999 then do; stopwk=991+ceil((stpdays+5)/7); end;
if stopy=2000 then do; stopwk=1043+ceil((stpdays+6)/7); end;
if stopy=2001 then do; stopwk=1096+ceil((stpdays+1)/7); end;
end;
if stopy<0 and stopy>-4 then do; stopwk=-3; end;
if -4<ostarty<0 then eyear=1;
end;

/* Account for backreporters */
do i=1 to 9;
  if backr=1 then do; startwk=.; stopwk=.; uid=-4; end;
end;

/* To create a seamless list of information from Round3 to Round4, the following lines of code are included. By
decreasing Round4 interview date by one, (which was done in emp_interview.sas) jobs that are worked up to
the interview date are also decreased by one. This is done so that the R4 interview week will not be counted
twice for respondents with a job during that time. The same procedure is used in the previous rounds. */
do over stopwk;
  if stopwk>0 and UID>0 then do;
    if stopwk>intwk then do; stopwk=intwk; end;
  end;
end;

/* The following lines consider jobs that begin the same week as the Round4 interview date. Since we are updating the
Round2 interview week by -1, we need to account for jobs that start in the same week or tenures of -1 will result. */
do over startwk;
  if startwk>0 and UID>0 then do;
    if startwk>intwk then do; startwk=intwk; end;
  end;
end;

/* The start date for self-employment jobs will be the week of Jan.1 of the year of the respondent's 18th birthday. This
method cuts off 3 jobs entirely (both the start and stop week are before Jan. 1 of the 18th birthday), and
shortens 19 other jobs. 20 respondents are affected. */
array bself (i) bself1-bself9;
array eself (i) eself1-eself9;
do over startwk;
  if self=1 and 0<startwk<(ag18jan-1) then do; bself=1; bcheck=1; end;
  if self=1 and 0<startwk<(ag18jan-1) and 0<stopwk<(ag18jan-1) then do; eself=1; echeck=1; end;

```

```

end;
do over startwk;
  if bself=1 then do; startwk=ag18jan; end;
  if eself=1 then do; startwk=.; stopwk=.; end;
end;

/* To account for Round 4 non-interview cases */
if starmo1=-5 then do;
  do over startwk; startwk=-5; stopwk=-5; end;
end;

/* Tests for missing start/stop data */
miss=0;
do i=1 to 9;
  if (startm=-4 and stopm>-4) or (startm>-4 and stopm=-4) or (startd=-4 and startd>-4) or (startd>-4 and stopd=-4) or
     (starty=-4 and starty>-4) or (starty>-4 and stopy=-4) then do; miss=1; end;
end;

/* Correcting for imputed values that resulted in the start date being later than the stop date. In these cases, the imputed
   date will be updated to the good date.*/
do i=1 to 9;
  if startwk>stopwk and (flag1=1 or flag2=1 or flag3=1) then do; startwk=stopwk; end;
  if startwk>stopwk and (flag4=1 or flag5=1 or flag6=1) then do; stopwk=startwk; end;
end;

/* Check that previously reported jobs do not predate the date of last interview */
do over UID;
  if UID<=200000 and startwk<r1int and (flag1=1 or flag2=1 or flag3=1) and startwk ne . and r3int=-5 and r2int=-5
     and r1int>0 then do; check=1; end;
  if UID<=200000 and startwk<r2int and (flag1=1 or flag2=1 or flag3=1) and startwk ne . and r3int=-5 and r2int>0
     then do; check=1; end;
  if UID<=200000 and startwk<r3int and (flag1=1 or flag2=1 or flag3=1) and startwk ne . and r3int>0 then do;
    check=1; end;
end;

/* Check for imputed start/stop date in incorrect chronological order */
do i=1 to 9;
  if startwk>stopwk and (flag1=1 or flag2=1 or flag3=1) and (flag4=1 or flag5=1 or flag6=1) then do; checky=1; end;
end;

/* Check for back-reporters, i.e. people that report a new job (UID>200000) that begins before the last interview date.*/
do over UID;
  if startwk<r3int and startwk>0 and r3int>0 then do; back3=1; back=1; end;
  if startwk<r2int and startwk>0 and r2int>0 and r3int=-5 then do; back2=1; back=1; end;
  if startwk<r1int and startwk>0 and r1int>0 and r2int=-5 and r3int=-5 then do; back1=1; back=1; end;
end;

/* Check that UID's are different at each loop, when greater than -4.*/
do i=2 to 9; if UID1=UID and UID1>-4 then umatch=1; end;
do i=3 to 9; if UID2=UID and UID2>-4 then umatch=1; end;
do i=4 to 9; if UID3=UID and UID3>-4 then umatch=1; end;
do i=5 to 9; if UID4=UID and UID4>-4 then umatch=1; end;
do i=6 to 9; if UID5=UID and UID5>-4 then umatch=1; end;
do i=7 to 9; if UID6=UID and UID6>-4 then umatch=1; end;
do i=8 to 9; if UID7=UID and UID7>-4 then umatch=1; end;
if UID8=UID9 and UID8>-4 then umatch=1;

```

```

/* Check that start/stop dates have positive UID values. */
do i=1 to 9;
  if startwk>0 and UID=-4 then umiss=1;
  if uid>0 and ostarty=-4 then smiss=1;
end;

/* Check to see how many start weeks for self-employed jobs come before the R's 18th birthday. */
array s18ck (i) s18ck1-s18ck9;
array e18ck (i) e18ck1-e18ck9;
do i=1 to 9;
  if startwk>0 and -5<age18wk<9999 and startwk<ag18jan and self=1 and smofl=0 then do; s18ck=1; slook=1; end;
  if stopwk>0 and -5<age18wk<9999 and stopwk<ag18jan and self=1 and emofl=0 then do; e18ck=1; elook=1; end;
end;

array diff (i) diff1-diff9;
do i=1 to 9;
  if s18ck=1 then do; diff=ag18jan-startwk; end;
end;

do i=1 to 9; if (self=1 and 0<uid<199999) then how=1; end;

/* Check for number of people older than 20 at dli (62 in R3, none in previous rounds) and has a job after their 20th
   birthday. (48 at R3). */
do i=1 to 9;
  if age20wk<r2int then do; why=1; end;
end;

***** Section 4: Convert dates of within-job gaps into NLSY97 week numbers. *****/
/** JOB 1 GAPS */
/* Another hand-edit for a data inconsistency. */
if id=1547241 then do; bgdy1_1=-4; bgmo1_1=-4; bgyr1_1=-4; egdy1_1=-4; egmo1_1=-4; egyr1_1=-4; end;

/* These variables are read as follows:
   BGDY1_1 = Begin day of within-job gap 1 on job 1
   EGMO1_5 = End month of within-job gap 5 on job 1
   BGAP1_3 = Begin week of within-job gap 3 on job 1 [CREATED] */
array bgdy (i) BGDY1_1-BGYD1_10; array bgmo (i) BGMO1_1-BGMO1_10; array bgyr (i) BGYR1_1-BGYR1_10;
array egdy (i) EGDY1_1-EGDY1_10; array egmo (i) EGMO1_1-EGMO1_10; array egyr (i) EGYR1_1-EGYR1_10;
array bdays (i) bday1_1-bday1_10; /* begin day of job1_gap# (internal calculation)*/
array edays (i) eday1_1-eday1_10; /* end day of job1_gap# (internal calculation)*/
array bweek (i) bgap1_1-bgap1_10; /* begin week of job1_gap# (created) */
array eweek (i) egap1_1-egap1_10; /* end week of job1_gap# (created) */
array bflag (i) bflg1_1-bflg1_10;
array eflag (i) eflg1_1-eflg1_10;
array bgfl (i) bgfl1_1-bgfl1_10;
array egfl (i) egfl1_1-egfl1_10;

/* only impute start/stop dates if day is missing */ /* Fill-in start day for those missing */
/* Missing gap days (start and stop) are now imputed to 1. Repeated without comment at each job. */
do over bgyr;
  if bgyr>0 then do;
    if bgmo>0 and bgdy<=0 then do; bgdy=1; bflag=1; end;
    /* Account for beginning gap dates before job start date.*/
    if bgyr=staryr1 and bgmo=starmo1 and bgdy<stardy1 and bflag=1 then do; bgdy=stardy1; end;
  end;
end;

```

```

/* Fill-in stop day for those missing */
do over egyr;
if egyr>0 then do;
  if egmo>0 and egdy<=0 then do; egdy=1; eflag=1;
  end;
  /* This accounts for end gap dates after job start date. */
  if egyr=stopyr1 and egmo=stopmo1 and egdy>stopdy1 and eflag=1 then do; egdy=stopdy1; end;
end;
end;

/*Set flag for gap exists but invalid data*/
do over bgyr;
  bgfl=-4;
  if (-4 < bgmo < 0) or (-4 < bgyr < 0) then do; bgfl=1; end;
end;
do over egyr;
  egfl=-4;
  if (-4 < egyr < 0) or (-4 < egmo < 0) then do; egfl=1; end;
end;

***Identify within-job gaps on JOB 1 ***;
***Convert gap dates to week numbers ***; ***Convert START month and day to total days (BDAYS);
do over bgmo;
  if bgmo>0 and bgdy>0 then do;
    if bgmo=1 then bdays=bgdy;
    if bgmo=3 then bdays=bgdy+59;
    if bgmo=5 then bdays=bgdy+120;
    if bgmo=7 then bdays=bgdy+181;
    if bgmo=9 then bdays=bgdy+243;
    if bgmo=11 then bdays=bgdy+304;
    if bgmo=2 then bdays=bgdy+31;
    if bgmo=4 then bdays=bgdy+90;
    if bgmo=6 then bdays=bgdy+151;
    if bgmo=8 then bdays=bgdy+212;
    if bgmo=10 then bdays=bgdy+273;
    if bgmo=12 then bdays=bgdy+334;
  end;
end;

***Account for leap years;
do over bgyr;
  if bgyr=1980 or bgyr=1984 or bgyr=1988 or bgyr=1992 or bgyr=1996 or bgyr=2000 then do;
    if bgmo>0 and bgdy>0 then do;
      if bgmo=1 then bdays=bgdy;
      if bgmo=3 then bdays=bgdy+60;
      if bgmo=5 then bdays=bgdy+121;
      if bgmo=7 then bdays=bgdy+182;
      if bgmo=9 then bdays=bgdy+244;
      if bgmo=11 then bdays=bgdy+305;
      if bgmo=2 then bdays=bgdy+31;
      if bgmo=4 then bdays=bgdy+91;
      if bgmo=6 then bdays=bgdy+152;
      if bgmo=8 then bdays=bgdy+213;
      if bgmo=10 then bdays=bgdy+274;
      if bgmo=12 then bdays=bgdy+335;
    end;
  end;
end;

***Convert STOP month and day to total days (EDAYS);
do over egmo;
  if egmo>0 and egdy>0 then do;
    if egmo=1 then edays=egdy;
    if egmo=3 then edays=egdy+59;
    if egmo=5 then edays=egdy+120;
    if egmo=7 then edays=egdy+181;
    if egmo=9 then edays=egdy+243;
    if egmo=11 then edays=egdy+304;
    if egmo=2 then edays=egdy+31;
    if egmo=4 then edays=egdy+90;
    if egmo=6 then edays=egdy+151;
    if egmo=8 then edays=egdy+212;
    if egmo=10 then edays=egdy+273;
    if egmo=12 then edays=egdy+334;
  end;
end;

```

Appendix 2: Employment Variable Creation

end;

***Account for leap years;

do over egyr;

if egyr=1980 or egyr=1984 or egyr=1988 or egyr=1992 or egyr=1996 or egyr=2000 then do;

if egmo>0 and egdy>0 then do;

if egmo=1 then edays=egdy;

if egmo=2 then edays=egdy+31;

if egmo=3 then edays=egdy+60;

if egmo=4 then edays=egdy+91;

if egmo=5 then edays=egdy+121;

if egmo=6 then edays=egdy+152;

if egmo=7 then edays=egdy+182;

if egmo=8 then edays=egdy+213;

if egmo=9 then edays=egdy+244;

if egmo=10 then edays=egdy+274;

if egmo=11 then edays=egdy+305;

if egmo=12 then edays=egdy+335;

end;

end;

end;

***Convert days into week numbers;

do over bgyr;

if bgyr>0 and bdays>0 then do;

if bgyr=1980 then do; bweek=ceil((bdays+2)/7); end;

if bgyr=1981 then do; bweek=52+ceil((bdays+4)/7); end;

if bgyr=1982 then do; bweek=104+ceil((bdays+5)/7); end;

if bgyr=1983 then do; bweek=156+ceil((bdays+6)/7); end;

if bgyr=1984 then do; bweek=209+ceil((bdays)/7); end;

if bgyr=1985 then do; bweek=261+ceil((bdays+2)/7); end;

if bgyr=1986 then do; bweek=313+ceil((bdays+3)/7); end;

if bgyr=1987 then do; bweek=365+ceil((bdays+4)/7); end;

if bgyr=1988 then do; bweek=417+ceil((bdays+5)/7); end;

if bgyr=1989 then do; bweek=470+ceil((bdays)/7); end;

if bgyr=1990 then do; bweek=522+ceil((bdays+1)/7); end;

if bgyr=1991 then do; bweek=574+ceil((bdays+2)/7); end;

if bgyr=1992 then do; bweek=626+ceil((bdays+3)/7); end;

if bgyr=1993 then do; bweek=678+ceil((bdays+5)/7); end;

if bgyr=1994 then do; bweek=730+ceil((bdays+6)/7); end;

if bgyr=1995 then do; bweek=783+ceil((bdays)/7); end;

if bgyr=1996 then do; bweek=835+ceil((bdays+1)/7); end;

if bgyr=1997 then do; bweek=887+ceil((bdays+3)/7); end;

if bgyr=1998 then do; bweek=939+ceil((bdays+4)/7); end;

if bgyr=1999 then do; bweek=991+ceil((bdays+5)/7); end;

if bgyr=2000 then do; bweek=1043+ceil((bdays+6)/7); end;

if bgyr=2001 then do; bweek=1096+ceil((bdays+1)/7); end;

if bweek>0 then do; bweek=bweek+1; end;

end;

end;

do over egyr;

if egyr>0 and edays>0 then do;

if egyr=1980 then do; eweek=ceil((edays+2)/7); end;

if egyr=1981 then do; eweek=52+ceil((edays+4)/7); end;

if egyr=1982 then do; eweek=104+ceil((edays+5)/7); end;

if egyr=1983 then do; eweek=156+ceil((edays+6)/7); end;

if egyr=1984 then do; eweek=209+ceil((edays)/7); end;

if egyr=1985 then do; eweek=261+ceil((edays+2)/7); end;

if egyr=1986 then do; eweek=313+ceil((edays+3)/7); end;

if egyr=1987 then do; eweek=365+ceil((edays+4)/7); end;

if egyr=1988 then do; eweek=417+ceil((edays+5)/7); end;

if egyr=1989 then do; eweek=470+ceil((edays)/7); end;

```

if egyr=1990 then do; eweek=522+ceil((edays+1)/7); end;
if egyr=1991 then do; eweek=574+ceil((edays+2)/7); end;
if egyr=1992 then do; eweek=626+ceil((edays+3)/7); end;
if egyr=1993 then do; eweek=678+ceil((edays+5)/7); end;
if egyr=1994 then do; eweek=730+ceil((edays+6)/7); end;
if egyr=1995 then do; eweek=783+ceil((edays)/7); end;
if egyr=1996 then do; eweek=835+ceil((edays+1)/7); end;
if egyr=1997 then do; eweek=887+ceil((edays+3)/7); end;
if egyr=1998 then do; eweek=939+ceil((edays+4)/7); end;
if egyr=1999 then do; eweek=991+ceil((edays+5)/7); end;
if egyr=2000 then do; eweek=1043+ceil((edays+6)/7); end;
if egyr=2001 then do; eweek=1096+ceil((edays+1)/7); end;
if eweek>0 then do; eweek=eweek-1; end;
end;
end;

array update (i) update1-update10;

/* The following lines omit gap start and stop dates for gaps less than one work week (5 days) */
do over bdays;
  if edays-bdays<5 and bweek>eweek and bdays ne . and edays ne . then do; bweek=.; eweek=.; end;
end;

/* The following lines are for end gap dates that exceed the Round4 interview date due to our rounding methods. */
do over eweek;
  if 0<eweek<1400 and eweek>intwk then do; eweek=intwk; end;
end;

/* The following omits cases where bweek>eweek, which are caused when missing values are substituted in. For
   example, when the day of a beginning gap is unknown, the program uses the 1st. This can cause bweek>eweek,
   which will cause problems when writing programs for the created variables. This situation will be fixed by
   making eweek and bweek the same. This is repeated (without comment) for each job. */
do over bweek;
  if 0<eweek<1200 then do;
    if bweek>eweek then do; bweek=eweek; update=1; end;
  end;
end;

/* If a respondent reports a gap on previous Round's job that occurred before the dli week, then we will not use this
   information. However, a newly reported job (i.e., 2000 UID) containing within job gaps that occurred before
   the dli week will be counted. This is repeated (without comment) for each job. Note that we have respondents
   who were not interviewed in previous round(s). The same process is done for these people, except we use the
   date of last interview. */
/* If a gap begins before the Round 3 interview date and ends after the Round 3 interview date, the beginning gap date
   is updated to the Round 3 interview date. This is repeated at each job without comment. */
do over bweek;
  if UID1>0 and dliwk>0 and bweek>0 and eweek>0 and backr1 ne 1 then do;
    /* Entire gap is before dli. */
    if dliwk>bweek and dliwk>eweek then do; bweek=.; eweek=.; end;
    /* B.gap is before dli, e.gap is after dli. */
    if dliwk<eweek and dliwk>bweek then do; bweek=dliwk; end;
  end;
end;

/* The following erases gaps that occur after the job has ended, only in cases where there are no imputed job start or
   stop months or years (imputed job start/stop days acceptable) */
do over bweek;

```

Appendix 2: Employment Variable Creation

```
if starw1>0 and stopw1>0 and smofl1=0 and emofl1=0 then do;
  if bweek>stopw1 then do; bweek=.; end;
  if eweek>stopw1 then do; eweek=.; end;
end;
end;

/* To correct for bad gap information.*/
do over bdays;
  if eweek>stopw1 and eweek ne . and stopw1>0 then eweek=stopw1;
  if bweek<starw1 and bweek ne . and bweek>0 then bweek=starw1;
  if bweek>stopw1 and bweek ne . and stopw1>0 then huh=1;
  if eweek<starw1 and eweek ne . and starw1>0 then huh=1;
end;

/*At this point the program loops through the same code for gaps on jobs 2-9. Due to space considerations, this code is
not repeated here. For more information please contact NLS User Services.*/

/* The following hand edits are unmatched cases, where two identical UID's are in the Round 4 roster. These are the same
jobs, and the following condenses them into one job with a within-job gap. */
if id=1216571 then do; starw1=1055; bgap1_1=1079; egap1_1=1086; stopw1=1095; starw2=.; stopw2=.; uid2=-4; end;
if id=1273771 then do; starw1=1075; bgap1_1=1078; egap1_1=1091; stopw1=1112; starw4=.; stopw4=.; uid4=-4; end;
if id=1278971 then do; starw1=1046; bgap1_1=1063; egap1_1=1080; stopw1=1096; bgap1_2=1095; egap1_2=1096;
  starw4=.; stopw4=.; uid4=-4; end;
if id=1329391 then do; starw1=1044; bgap1_1=1047; egap1_1=1067; stopw1=1079; starw3=.; stopw3=.; uid3=-4; end;
if id=1417821 then do; starw3=1041; bgap3_1=1050; egap3_1=1064; stopw3=1072; starw4=.; stopw4=.; uid4=-4; end;

***** Section 5: Weeks worked by respondent on each job *****
/* This section counts the weeks worked by the respondent and removes the within job gaps. It will place a "1" into
weeks where the respondent was employed. Weeks range from the first week of 1980 to the last week of 2001,
for a total of 1148 weeks. */

array job1wks (i) wk1_1-wk1_1148;           array job2wks (i) wk2_1-wk2_1148;
array job3wks (i) wk3_1-wk3_1148;           array job4wks (i) wk4_1-wk4_1148;
array job5wks (i) wk5_1-wk5_1148;           array job6wks (i) wk6_1-wk6_1148;
array job7wks (i) wk7_1-wk7_1148;           array job8wks (i) wk8_1-wk8_1148;
array job9wks (i) wk9_1-wk9_1148;           array stopw (i) stopw1-stopw9;

* Default Settings;
do i=1 to 1148;
  job1wks=0;      job2wks=0;      job3wks=0;      job4wks=0;      job5wks=0;      job6wks=0;
  job7wks=0;      job8wks=0;      job9wks=0;
end;

* Change military start and stop dates to missing since only civilian jobs are counted;
if milflag1=1 then do; starw1=.; stopw1=.; end;
if milflag2=1 then do; starw2=.; stopw2=.; end;
if milflag3=1 then do; starw3=.; stopw3=.; end;
if milflag4=1 then do; starw4=.; stopw4=.; end;
if milflag5=1 then do; starw5=.; stopw5=.; end;
if milflag6=1 then do; starw6=.; stopw6=.; end;
if milflag7=1 then do; starw7=.; stopw7=.; end;
if milflag8=1 then do; starw8=.; stopw8=.; end;
if milflag9=1 then do; starw9=.; stopw9=.; end;

/* Define rd4wk as the maximum of dliwk and age14wk. This is used for bad start/stop weeks. For the re-interview
cases, rd3wk will be set to the maximum of the Round 1 interview date and the respondent's 14th birthday. */
```

```

if dliwk>0 then do;
  if dliwk>age14wk then do; rd3wk=dliwk; end;
  if age14wk=>dliwk then do; rd3wk=age14wk; end;
end;

/** TOTAL WEEKS WORKED ON JOB 1 **/
starfl_1=0; stopfl_1=0;

if starw1=-3 and uid1 ne -5 then do; starw1=rd3wk; starfl_1=1; end;
if stopw1=-3 and uid1 ne -5 then do; stopw1=intwk; stopfl_1=1; end;
if smofl1=1 then do; starfl_1=1; end;
if emofl1=1 then do; stopfl_1=1; end;

if starw1>0 and stopw1>0 then do;
  do i=(starw1) to (stopw1); job1wks=1; end;
  *** Remove gap 1 on job 1;
  if bgap1_1>0 & egap1_1>0 then do; do i=(bgap1_1) to (egap1_1); job1wks=0; end; end;
  *** Remove gap 2 on job 1;
  if bgap1_2>0 & egap1_2>0 then do; do i=(bgap1_2) to (egap1_2); job1wks=0; end; end;
  *** Remove gap 3 on job 1;
  if bgap1_3>0 & egap1_3>0 then do; do i=(bgap1_3) to (egap1_3); job1wks=0; end; end;
  *** Remove gap 4 on job 1;
  if bgap1_4>0 & egap1_4>0 then do; do i=(bgap1_4) to (egap1_4); job1wks=0; end; end;
  *** Remove gap 5 on job 1;
  if bgap1_5>0 & egap1_5>0 then do; do i=(bgap1_5) to (egap1_5); job1wks=0; end; end;
  *** Remove gap 6 on job 1;
  if bgap1_6>0 & egap1_6>0 then do; do i=(bgap1_6) to (egap1_6); job1wks=0; end; end;
  *** Remove gap 7 on job 1;
  if bgap1_7>0 & egap1_7>0 then do; do i=(bgap1_7) to (egap1_7); job1wks=0; end; end;
  *** Remove gap 8 on job 1;
  if bgap1_8>0 & egap1_8>0 then do; do i=(bgap1_8) to (egap1_8); job1wks=0; end; end;
  *** Remove gap 9 on job 1;
  if bgap1_9>0 & egap1_9>0 then do; do i=(bgap1_9) to (egap1_9); job1wks=0; end; end;
  *** Remove gap 10 on job 1;
  if bgap1_10>0 & egap1_10>0 then do; do i=(bgap1_10) to (egap1_10); job1wks=0; end; end;
  *** Remove gap 1 on job 1 - beginning gap date bad;
  if bgfl1_1=1 & egap1_1>0 then do; do i=(starw1) to (egap1_1); job1wks=-3; gpfl1_1=1; end; end;
  *** Remove gap 1 on job 1 - end gap date bad;
  if bgap1_1>0 & egf1_1=1 then do; do i=(bgap1_1) to min(stopw1, bgap1_2-1, bgap1_3-1, bgap1_4-1, bgap1_5-1,
    bgap1_6-1, bgap1_7-1, bgap1_8-1, bgap1_9-1, bgap1_10-1, egap1_2+1, egap1_3+1, egap1_4+1, egap1_5+1,
    egap1_6+1, egap1_7+1, egap1_8+1, egap1_9+1, egap1_10+1); job1wks=-3; gpfl1_1=1; end; end;
  *** Remove gap 1 on job 1 - both gap dates bad;
  if bgfl1_1=1 & egf1_1=1 then do; do i=(starw1) to min(stopw1, bgap1_2-1, bgap1_3-1, bgap1_4-1, bgap1_5-1,
    bgap1_6-1, bgap1_7-1, bgap1_8-1, bgap1_9-1, bgap1_10-1, egap1_2+1, egap1_3+1, egap1_4+1, egap1_5+1,
    egap1_6+1, egap1_7+1, egap1_8+1, egap1_9+1, egap1_10+1); job1wks=-3; gpfl1_1=1; end; end;
  *** Remove gap 2 on job 1 - beginning gap date bad;
  if bgfl1_2=1 & egap1_2>0 then do; do i=max(starw1,bgap1_1-1,egap1_1+1) to (egap1_2); job1wks=-3; gpfl1_2=1;
    end; end;
  *** Remove gap 2 on job 1 - end gap date bad;
  if bgap1_2>0 & egf1_2=1 then do; do i=(bgap1_2) to min(stopw1, bgap1_3-1, bgap1_4-1, bgap1_5-1, bgap1_6-1,
    bgap1_7-1, bgap1_8-1, bgap1_9-1, bgap1_10-1, egap1_3+1, egap1_4+1, egap1_5+1, egap1_6+1, egap1_7+1,
    egap1_8+1, egap1_9+1, egap1_10+1); job1wks=-3; gpfl1_2=1; end; end;
  *** Remove gap 2 on job 1 - both gap dates bad;
  if bgfl1_2=1 & egf1_2=1 then do; do i=max(starw1,bgap1_1-1,egap1_1+1) to min(stopw1, bgap1_3-1, bgap1_4-1,
    bgap1_5-1, bgap1_6-1, bgap1_7-1, bgap1_8-1, bgap1_9-1, bgap1_10-1, egap1_3+1, egap1_4+1, egap1_5+1,
    egap1_6+1, egap1_7+1, egap1_8+1, egap1_9+1, egap1_10+1); job1wks=-3; gpfl1_2=1; end; end;
  *** Remove gap 3 on job 1 - beginning gap date bad;

```

Appendix 2: Employment Variable Creation

```

if bgfl1_3=1 & egap1_3>0 then do; do i=max(starw1,bgap1_1-1,bgap1_2-1,egap1_1+1,egap1_2+1) to (egap1_3);
    job1wks=-3; gpfl1_3=1; end; end;
*** Remove gap 3 on job 1 - end gap date bad;
if bgap1_3>0 & egfl1_3=1 then do; do i=(bgap1_3) to min(stopw1, bgap1_4-1, bgap1_5-1, bgap1_6-1, bgap1_7-1,
    bgap1_8-1, bgap1_9-1, bgap1_10-1, egap1_4+1, egap1_5+1, egap1_6+1, egap1_7+1, egap1_8+1, egap1_9+1,
    egap1_10+1); job1wks=-3; gpfl1_3=1; end; end;
*** Remove gap 3 on job 1 - both gap dates bad;
if bgfl1_3=1 & egfl1_3=1 then do; do i=max(starw1, bgap1_1-1, bgap1_2-1, egap1_1+1, egap1_2+1) to min(stopw1,
    bgap1_4-1, bgap1_5-1, bgap1_6-1, bgap1_7-1, bgap1_8-1, bgap1_9-1, bgap1_10-1, egap1_4+1, egap1_5+1,
    egap1_6+1, egap1_7+1, egap1_8+1, egap1_9+1, egap1_10+1); job1wks=-3; gpfl1_3=1; end; end;
*** Remove gap 4 on job 1 - beginning gap date bad;
if bgfl1_4=1 & egap1_4>0 then do; do i=max(starw1, bgap1_1-1, bgap1_2-1, bgap1_3-1, egap1_1+1, egap1_2+1,
    egap1_3+1) to (egap1_4); job1wks=-3; gpfl1_4=1; end; end;
*** Remove gap 4 on job 1 - end gap date bad;
if bgap1_4>0 & egfl1_4=1 then do; do i=(bgap1_4) to min(stopw1, bgap1_5-1, bgap1_6-1, bgap1_7-1, bgap1_8-1,
    bgap1_9-1, bgap1_10-1, egap1_5+1, egap1_6+1, egap1_7+1, egap1_8+1, egap1_9+1, egap1_10+1);
    job1wks=-3; gpfl1_4=1; end; end;
*** Remove gap 4 on job 1 - both gap dates bad;
if bgfl1_4=1 & egfl1_4=1 then do; do i=max(starw1, bgap1_1-1, bgap1_2-1, bgap1_3-1, egap1_1+1, egap1_2+1,
    egap1_3+1) to min(stopw1, bgap1_5-1, bgap1_6-1, bgap1_7-1, bgap1_8-1, bgap1_9-1, bgap1_10-1,
    egap1_5+1, egap1_6+1, egap1_7+1, egap1_8+1, egap1_9+1, egap1_10+1); job1wks=-3; gpfl1_4=1; end; end;
*** Remove gap 5 on job 1 - beginning gap date bad;
if bgfl1_5=1 & egap1_5>0 then do; do i=max(starw1, bgap1_1-1, bgap1_2-1, bgap1_3-1, bgap1_4-1, egap1_1+1,
    egap1_2+1, egap1_3+1, egap1_4+1) to (egap1_5); job1wks=-3; gpfl1_5=1; end; end;
*** Remove gap 5 on job 1 - end gap date bad;
if bgap1_5>0 & egfl1_5=1 then do; do i=(bgap1_5) to min(stopw1, bgap1_6-1, bgap1_7-1, bgap1_8-1, bgap1_9-1,
    bgap1_10-1, egap1_6+1, egap1_7+1, egap1_8+1, egap1_9+1, egap1_10+1); job1wks=-3; gpfl1_5=1; end; end;
*** Remove gap 5 on job 1 - both gap dates bad;
if bgfl1_5=1 & egfl1_5=1 then do; do i=max(starw1, bgap1_1-1, bgap1_2-1, bgap1_3-1, bgap1_4-1, egap1_1+1,
    egap1_2+1, egap1_3+1, egap1_4+1) to min(stopw1, bgap1_6-1, bgap1_7-1, bgap1_8-1, bgap1_9-1,
    bgap1_10-1, egap1_6+1, egap1_7+1, egap1_8+1, egap1_9+1, egap1_10+1); job1wks=-3; gpfl1_5=1; end; end;
*** Remove gap 6 on job 1 - beginning gap date bad;
if bgfl1_6=1 & egap1_6>0 then do; do i=max(starw1, bgap1_1-1, bgap1_2-1, bgap1_3-1, bgap1_4-1, bgap1_5-1,
    egap1_1+1, egap1_2+1, egap1_3+1, egap1_4+1, egap1_5+1) to (egap1_6); job1wks=-3; gpfl1_6=1; end; end;
*** Remove gap 6 on job 1 - end gap date bad;
if bgap1_6>0 & egfl1_6=1 then do; do i=(bgap1_6) to min(stopw1, bgap1_7-1, bgap1_8-1, bgap1_9-1, bgap1_10-1,
    egap1_7+1, egap1_8+1, egap1_9+1, egap1_10+1); job1wks=-3; gpfl1_6=1; end; end;
*** Remove gap 6 on job 1 - both gap dates bad;
if bgfl1_6=1 & egfl1_6=1 then do; do i=max(starw1, bgap1_1-1, bgap1_2-1, bgap1_3-1, bgap1_4-1, bgap1_5-1,
    egap1_1+1, egap1_2+1, egap1_3+1, egap1_4+1, egap1_5+1) to min(stopw1, bgap1_6-1, bgap1_7-1,
    bgap1_8-1, bgap1_9-1, bgap1_10-1, egap1_6+1, egap1_7+1, egap1_8+1, egap1_9+1, egap1_10+1);
    job1wks=-3; gpfl1_6=1; end; end;
*** Remove gap 7 on job 1 - beginning gap date bad;
if bgfl1_7=1 & egap1_7>0 then do; do i=max(starw1, bgap1_1-1, bgap1_2-1, bgap1_3-1, bgap1_4-1, bgap1_5-1,
    bgap1_6-1, egap1_1+1, egap1_2+1, egap1_3+1, egap1_4+1, egap1_5+1, egap1_6+1) to (egap1_7);
    job1wks=-3; gpfl1_7=1; end; end;
*** Remove gap 7 on job 1 - end gap date bad;
if bgap1_7>0 & egfl1_7=1 then do; do i=(bgap1_7) to min(stopw1, bgap1_8-1, bgap1_9-1, bgap1_10-1, egap1_8+1,
    egap1_9+1, egap1_10+1); job1wks=-3; gpfl1_7=1; end; end;
*** Remove gap 7 on job 1 - both gap dates bad;
if bgfl1_7=1 & egfl1_7=1 then do; do i=max(starw1, bgap1_1-1, bgap1_2-1, bgap1_3-1, bgap1_4-1, bgap1_5-1,
    bgap1_6-1, egap1_1+1, egap1_2+1, egap1_3+1, egap1_4+1, egap1_5+1, egap1_6+1) to min(stopw1,
    bgap1_8-1, bgap1_9-1, bgap1_10-1, egap1_8+1, egap1_9+1, egap1_10+1); job1wks=-3; gpfl1_7=1; end; end;
*** Remove gap 8 on job 1 - beginning gap date bad;
if bgfl1_8=1 & egap1_8>0 then do; do i=max(starw1, bgap1_1-1, bgap1_2-1, bgap1_3-1, bgap1_4-1, bgap1_5-1,
    bgap1_6-1, bgap1_7-1, egap1_1+1, egap1_2+1, egap1_3+1, egap1_4+1, egap1_5+1, egap1_6+1, egap1_7+1)
    to (egap1_8); job1wks=-3; gpfl1_8=1; end; end;

```

```

*** Remove gap 8 on job 1 - end gap date bad;
if bgap1_8>0 & egf1_8=1 then do; do i=(bgap1_8) to min(stopw1, bgap1_9-1, bgap1_10-1, egap1_9+1,
    egap1_10+1); job1wks=-3; gpfl1_8=1; end; end;
*** Remove gap 8 on job 1 - both gap dates bad;
if bgf1_8=1 & egf1_8=1 then do; do i=max(starw1, bgap1_1-1, bgap1_2-1, bgap1_3-1, bgap1_4-1, bgap1_5-1,
    bgap1_6-1, bgap1_7-1, egap1_1+1, egap1_2+1, egap1_3+1, egap1_4+1, egap1_5+1, egap1_6+1, egap1_7+1)
    to min(stopw1, bgap1_9-1, bgap1_10-1, egap1_9+1, egap1_10+1); job1wks=-3; gpfl1_8=1; end; end;
*** Remove gap 9 on job 1 - beginning gap date bad;
if bgf1_9=1 & egap1_9>0 then do; do i=max(starw1, bgap1_1-1, bgap1_2-1, bgap1_3-1, bgap1_4-1, bgap1_5-1,
    bgap1_6-1, bgap1_7-1, bgap1_8-1, egap1_1+1, egap1_2+1, egap1_3+1, egap1_4+1, egap1_5+1, egap1_6+1,
    egap1_7+1, egap1_8+1) to (egap1_9); job1wks=-3; gpfl1_9=1; end; end;
*** Remove gap 9 on job 1 - end gap date bad;
if bgap1_9>0 & egf1_9=1 then do; do i=(bgap1_9) to min(stopw1, bgap1_10-1, egap1_10+1); job1wks=-3;
    gpfl1_9=1; end; end;
*** Remove gap 9 on job 1 - both gap dates bad;
if bgf1_9=1 & egf1_9=1 then do; do i=max(starw1, bgap1_1-1, bgap1_2-1, bgap1_3-1, bgap1_4-1, bgap1_5-1,
    bgap1_6-1, bgap1_7-1, bgap1_8-1, egap1_1+1, egap1_2+1, egap1_3+1, egap1_4+1, egap1_5+1, egap1_6+1,
    egap1_7+1, egap1_8+1) to min(stopw1, bgap1_10-1, egap1_10+1); job1wks=-3; gpfl1_9=1; end; end;
*** Remove gap 10 on job 1 - beginning gap date bad;
if bgf1_10=1 & egap1_10>0 then do; do i=max(starw1, bgap1_1-1, bgap1_2-1, bgap1_3-1, bgap1_4-1, bgap1_5-1,
    bgap1_6-1, bgap1_7-1, bgap1_8-1, bgap1_9-1, egap1_1+1, egap1_2+1, egap1_3+1, egap1_4+1, egap1_5+1,
    egap1_6+1, egap1_7+1, egap1_8+1, egap1_9+1) to (egap1_10); job1wks=-3; gpfl1_10=1; end; end;
*** Remove gap 10 on job 1 - end gap date bad;
if bgap1_10>0 & egf1_10=1 then do; do i=(bgap1_10) to (stopw1); job1wks=-3; gpfl1_10=1; end; end;
*** Remove gap 10 on job 1 - both gap dates bad;
if bgf1_10=1 & egf1_10=1 then do; do i=max(starw1, bgap1_1-1, bgap1_2-1, bgap1_3-1, bgap1_4-1, bgap1_5-1,
    bgap1_6-1, bgap1_7-1, bgap1_8-1, bgap1_9-1, egap1_1+1, egap1_2+1, egap1_3+1, egap1_4+1, egap1_5+1,
    egap1_6+1, egap1_7+1, egap1_8+1, egap1_9+1) to (stopw1); job1wks=-3; gpfl1_10=1; end; end;
end;

if starfl_1=1 then do;
    do i=(starw1) to min(stopw1, bgap1_1-1, bgap1_2-1, bgap1_3-1, bgap1_4-1, bgap1_5-1, bgap1_6-1, bgap1_7-1,
        bgap1_8-1, bgap1_9-1, bgap1_10-1, egap1_1+1, egap1_2+1, egap1_3+1, egap1_4+1, egap1_5+1, egap1_6+1,
        egap1_7+1, egap1_8+1, egap1_9+1, egap1_10+1); job1wks=-3; end;
end;

if stopfl_1=1 then do;
    do i=max(stopw1, bgap1_1-1, bgap1_2-1, bgap1_3-1, bgap1_4-1, bgap1_5-1, bgap1_6-1, bgap1_7-1, bgap1_8-1,
        bgap1_9-1, bgap1_10-1, egap1_1+1, egap1_2+1, egap1_3+1, egap1_4+1, egap1_5+1, egap1_6+1,
        egap1_7+1, egap1_8+1, egap1_9+1, egap1_10+1) to (stopw1); job1wks=-3; end;
end;

/*At this point the program repeats the above code for jobs 2-9. This code is omitted due to space considerations. For
more information, please contact NLS User Services.*/
endsas;

```

NUMBER OF WEEKS WORKED DURING 19XX/20XX

Variables Created: CV_WKSWK_YR_ALL.80 – CV_WKSWK_YR_ALL.01
 CV_WKSWK_YR_ET.80 – CV_WKSWK_YR_ET.01
 CV_WKSWK_YR_SE.98 – CV_WKSWK_YR_SE.01

Programs Used

This program uses emp_begin.sas as input (see the first page of this appendix for details).

This program counts the number of weeks each respondent worked at any job for each year of potential work activity (1980-2001). Three sets of variables are created: one for all jobs, one for employee-type jobs, and one for self-employed jobs. See the introduction to this appendix for important information about the universes for each of these variables. Respondents not working in a given year are given a default value of zero (0) weeks worked. Otherwise, the variable indicates the actual cumulative weeks worked on all jobs in that year.

```
/* Section 1: Create variables for # of weeks worked at any
job during a given year based on round 3 data. */

array job1wks (i) wk1_1-wk1_1148;
array job2wks (i) wk2_1-wk2_1148;
array job3wks (i) wk3_1-wk3_1148;
array job4wks (i) wk4_1-wk4_1148;
array job5wks (i) wk5_1-wk5_1148;
array job6wks (i) wk6_1-wk6_1148;
array job7wks (i) wk7_1-wk7_1148;
array job8wks (i) wk8_1-wk8_1148;
array job9wks (i) wk9_1-wk9_1148;
array alljobs (i) wks1-wks1148;
array selfjobs (i) swks1-swks1148;
array empjobs (i) ewks1-ewks1148;
array starw (i) starw1-starw9;
array stopw (i) stopw1-stopw9;
array starfl (i) starfl_1-starfl_9;
array stopfl (i) stopfl_1-stopfl_9;
array uid (i) uid1-uid9;
array self (i) self1-self9;
array r4wks (i) r4wks80-r4wks99 r4wks00 r4wks01;
array r4swks (i) r4swks80-r4swks99 r4swks00 r4swks01;
array r4ewks (i) r4ewks80-r4ewks99 r4ewks00 r4ewks01;

/* Create a separate array for self employed jobs and
employee jobs for the new list of created variables due in
Round 4. The arrays are laid down in the same manner as
"alljobs," with the self-employed dummy deciding which
jobs are laid in each array. */
do over alljobs;
    alljobs=0; selfjobs=0; empjobs=0;
end;

/** Overlay multiple jobs over JOB 1 work weeks      **/ 

do over job1wks;
    alljobs=job1wks;
    if self1=1 then do; selfjobs=job1wks; end;
    if self1=0 then do; empjobs=job1wks; end;
end;

do over alljobs;
    if job2wks=1 then do; alljobs=job2wks; end;
    if job2wks=1 and self2=1 then do; selfjobs=job2wks;
        end;
    if job2wks=1 and self2=0 then do; empjobs=job2wks;
        end;
    if job2wks=-3 and alljobs=0 then do; alljobs=job2wks;
        end;
    if job2wks=-3 and selfjobs=0 and self2=1 then do;
        selfjobs=job2wks; end;
    if job2wks=-3 and empjobs=0 and self2=0 then do;
        empjobs=job2wks; end;
end;

/* At this point the program loops through the job2 code
for jobs 3-9. This code is omitted here. Contact
NLS User Services for more information.*/

/**Calculate cumulative weeks on all jobs for each year */
/* Initialize each annual c.v. */
do over r4wks;
    r4wks=0; r4swks=0; r4ewks=0;
end;

/* Need separate do loops for valid data ("1s") and for
invalid data ("-3s") or it will just add them up */
/* 1980 */
do i=1 to 52;
    if alljobs=1 then do; r4wks80=r4wks80+1; end;
    if selfjobs=1 then do; r4swks80=r4swks80+1; end;
    if empjobs=1 then do; r4ewks80=r4ewks80+1; end;
    if alljobs=-3 then do; r4wks80=-3; end;
    if selfjobs=-3 then do; r4swks80=-3; end;
    if empjobs=-3 then do; r4ewks80=-3; end;
end;

*****At this point the program loops through the
same code used above for 1980 for each year 1981-2001,
creating the variables r4wks81, r4swks81, and r4ewks81;
r4wks81, r4swks81, and r4ewks81; and so on through
```

r4wks01, r4swks01, and r4ewks01. These loops are deleted due to space considerations; users who need the entire code should contact NLS User Services. The week numbers for the "do i" statement for each year are as follows:

1981	53-104	1992	627-678
1982	105-156	1993	679-730
1983	157-209	1994	731-783
1984	210-261	1995	784-835
1985	262-313	1996	836-887
1986	314-365	1997	888-939
1987	366-417	1998	940-991
1988	418-470	1999	992-1044
1989	471-522	2000	1045-1096
1990	523-574	2001	1097-1148***/
1991	575-626		

```

do i=1 to 9;
/* start date invalid */
if starfl=1 then do;
  if stopw>1 and self=1 and r4wks80>0 then do;
    r4wks80=-3; r4swks80=-3; end;
  if stopw>1 and self=0 and r4wks80>0 then do;
    r4wks80=-3; r4ewks80=-3; end;
  if stopw>52 and self=1 and r4wks81>0 then do;
    r4wks81=-3; r4swks81=-3; end;
  if stopw>52 and self=0 and r4wks81>0 then do;
    r4wks81=-3; r4ewks81=-3; end;
  if stopw>104 and self=1 and r4wks82>0 then do;
    r4wks82=-3; r4swks82=-3; end;
  if stopw>104 and self=0 and r4wks82>0 then do;
    r4wks82=-3; r4ewks82=-3; end;
  if stopw>156 and self=1 and r4wks83>0 then do;
    r4wks83=-3; r4swks83=-3; end;
  if stopw>156 and self=0 and r4wks83>0 then do;
    r4wks83=-3; r4ewks83=-3; end;
  if stopw>209 and self=1 and r4wks84>0 then do;
    r4wks84=-3; r4swks84=-3; end;
  if stopw>209 and self=0 and r4wks84>0 then do;
    r4wks84=-3; r4ewks4=-3; end;
  if stopw>261 and self=1 and r4wks85>0 then do;
    r4wks85=-3; r4swks85=-3; end;
  if stopw>261 and self=0 and r4wks85>0 then do;
    r4wks85=-3; r4ewks85=-3; end;
  if stopw>313 and self=1 and r4wks86>0 then do;
    r4wks86=-3; r4swks86=-3; end;
  if stopw>313 and self=0 and r4wks86>0 then do;
    r4wks86=-3; r4ewks86=-3; end;
  if stopw>365 and self=1 and r4wks87>0 then do;
    r4wks87=-3; r4swks87=-3; end;
  if stopw>365 and self=0 and r4wks87>0 then do;
    r4wks87=-3; r4ewks87=-3; end;
  if stopw>417 and self=1 and r4wks88>0 then do;
    r4wks88=-3; r4swks88=-3; end;
  if stopw>417 and self=0 and r4wks88>0 then do;
    r4wks88=-3; r4ewks88=-3; end;

```

```

if stopw>470 and self=1 and r4wks89>0 then do;
  r4wks89=-3; r4swks89=-3; end;
if stopw>470 and self=0 and r4wks89>0 then do;
  r4wks89=-3; r4ewks89=-3; end;
if stopw>522 and self=1 and r4wks90>0 then do;
  r4wks90=-3; r4swks90=-3; end;
if stopw>522 and self=0 and r4wks90>0 then do;
  r4wks90=-3; r4ewks90=-3; end;
if stopw>574 and self=1 and r4wks91>0 then do;
  r4wks91=-3; r4swks91=-3; end;
if stopw>574 and self=0 and r4wks91>0 then do;
  r4wks91=-3; r4ewks91=-3; end;
if stopw>626 and self=1 and r4wks92>0 then do;
  r4wks92=-3; r4swks92=-3; end;
if stopw>626 and self=0 and r4wks92>0 then do;
  r4wks92=-3; r4ewks92=-3; end;
if stopw>678 and self=1 and r4wks93>0 then do;
  r4wks93=-3; r4swks93=-3; end;
if stopw>678 and self=0 and r4wks93>0 then do;
  r4wks93=-3; r4ewks93=-3; end;
if stopw>730 and self=1 and r4wks94>0 then do;
  r4wks94=-3; r4swks94=-3; end;
if stopw>730 and self=0 and r4wks94>0 then do;
  r4wks94=-3; r4ewks94=-3; end;
if stopw>783 and self=1 and r4wks95>0 then do;
  r4wks95=-3; r4swks95=-3; end;
if stopw>783 and self=0 and r4wks95>0 then do;
  r4wks95=-3; r4ewks95=-3; end;
if stopw>834 and self=1 and r4wks96>0 then do;
  r4wks96=-3; r4swks96=-3; end;
if stopw>834 and self=0 and r4wks96>0 then do;
  r4wks96=-3; r4ewks96=-3; end;
if stopw>887 and self=1 and r4wks97>0 then do;
  r4wks97=-3; r4swks97=-3; end;
if stopw>887 and self=0 and r4wks97>0 then do;
  r4wks97=-3; r4ewks97=-3; end;
if stopw>939 and self=1 and r4wks98>0 then do;
  r4wks98=-3; r4swks98=-3; end;
if stopw>939 and self=0 and r4wks98>0 then do;
  r4wks98=-3; r4ewks98=-3; end;
if stopw>990 and self=1 and r4wks99>0 then do;
  r4wks99=-3; r4swks99=-3; end;
if stopw>990 and self=0 and r4wks99>0 then do;
  r4wks99=-3; r4ewks99=-3; end;
if stopw>1043 and self=1 and r4wks00>0 then do;
  r4wks00=-3; r4swks00=-3; end;
if stopw>1043 and self=0 and r4wks00>0 then do;
  r4wks00=-3; r4ewks00=-3; end;
if stopw>1096 and self=1 and r4wks01>0 then do;
  r4wks01=-3; r4swks01=-3; end;
if stopw>1096 and self=0 and r4wks01>0 then do;
  r4wks01=-3; r4ewks01=-3; end;
end;

/* stop date invalid */
if stopfl=1 then do;

```

```

if starw<53 and self=1 and r4wks80>0 then do;
  r4wks80=-3; r4swks80=-3; end;
if starw<53 and self=0 and r4wks80>0 then do;
  r4wks80=-3; r4ewks80=-3; end;
if starw<105 and self=1 and r4wks81>0 then do;
  r4wks81=-3; r4swks81=-3; end;
if starw<105 and self=0 and r4wks81>0 then do;
  r4wks81=-3; r4ewks81=-3; end;
if starw<157 and self=1 and r4wks82>0 then do;
  r4wks82=-3; r4swks82=-3; end;
if starw<157 and self=0 and r4wks82>0 then do;
  r4wks82=-3; r4ewks82=-3; end;
if starw<210 and self=1 and r4wks83>0 then do;
  r4wks83=-3; r4swks83=-3; end;
if starw<210 and self=0 and r4wks83>0 then do;
  r4wks83=-3; r4ewks83=-3; end;
if starw<262 and self=1 and r4wks84>0 then do;
  r4wks84=-3; r4swks84=-3; end;
if starw<262 and self=0 and r4wks84>0 then do;
  r4wks84=-3; r4ewks84=-3; end;
if starw<314 and self=1 and r4wks85>0 then do;
  r4wks85=-3; r4swks85=-3; end;
if starw<314 and self=0 and r4wks85>0 then do;
  r4wks85=-3; r4ewks85=-3; end;
if starw<366 and self=1 and r4wks86>0 then do;
  r4wks86=-3; r4swks86=-3; end;
if starw<366 and self=0 and r4wks86>0 then do;
  r4wks86=-3; r4ewks86=-3; end;
if starw<418 and self=1 and r4wks87>0 then do;
  r4wks87=-3; r4swks87=-3; end;
if starw<418 and self=0 and r4wks87>0 then do;
  r4wks87=-3; r4ewks87=-3; end;
if starw<471 and self=1 and r4wks88>0 then do;
  r4wks88=-3; r4swks88=-3; end;
if starw<471 and self=0 and r4wks88>0 then do;
  r4wks88=-3; r4ewks88=-3; end;
if starw<523 and self=1 and r4wks89>0 then do;
  r4wks89=-3; r4swks89=-3; end;
if starw<523 and self=0 and r4wks89>0 then do;
  r4wks89=-3; r4ewks89=-3; end;
if starw<575 and self=1 and r4wks90>0 then do;
  r4wks90=-3; r4swks90=-3; end;
if starw<575 and self=0 and r4wks90>0 then do;
  r4wks90=-3; r4ewks90=-3; end;
if starw<627 and self=1 and r4wks91>0 then do;
  r4wks91=-3; r4swks91=-3; end;
if starw<627 and self=0 and r4wks91>0 then do;
  r4wks91=-3; r4ewks91=-3; end;
if starw<679 and self=1 and r4wks92>0 then do;
  r4wks92=-3; r4swks92=-3; end;
if starw<679 and self=0 and r4wks92>0 then do;
  r4wks92=-3; r4ewks92=-3; end;
if starw<731 and self=1 and r4wks93>0 then do;
  r4wks93=-3; r4swks93=-3; end;
if starw<731 and self=0 and r4wks93>0 then do;
  r4wks93=-3; r4ewks93=-3; end;

if starw<784 and self=1 and r4wks94>0 then do;
  r4wks94=-3; r4swks94=-3; end;
if starw<784 and self=0 and r4wks94>0 then do;
  r4wks94=-3; r4ewks94=-3; end;
if starw<836 and self=1 and r4wks95>0 then do;
  r4wks95=-3; r4swks95=-3; end;
if starw<836 and self=0 and r4wks95>0 then do;
  r4wks95=-3; r4ewks95=-3; end;
if starw<888 and self=1 and r4wks96>0 then do;
  r4wks96=-3; r4swks96=-3; end;
if starw<888 and self=0 and r4wks96>0 then do;
  r4wks96=-3; r4ewks96=-3; end;
if starw<940 and self=1 and r4wks97>0 then do;
  r4wks97=-3; r4swks97=-3; end;
if starw<940 and self=0 and r4wks97>0 then do;
  r4wks97=-3; r4ewks97=-3; end;
if starw<991 and self=1 and r4wks98>0 then do;
  r4wks98=-3; r4swks98=-3; end;
if starw<991 and self=0 and r4wks98>0 then do;
  r4wks98=-3; r4ewks98=-3; end;
if starw<1044 and self=1 and r4wks99>0 then do;
  r4wks99=-3; r4swks99=-3; end;
if starw<1044 and self=0 and r4wks99>0 then do;
  r4wks99=-3; r4ewks99=-3; end;
if starw<1096 and self=1 and r4wks00>0 then do;
  r4wks00=-3; r4swks00=-3; end;
if starw<1096 and self=0 and r4wks00>0 then do;
  r4wks00=-3; r4ewks00=-3; end;
if starw<1149 and self=1 and r4wks01>0 then do;
  r4wks01=-3; r4swks01=-3; end;
if starw<1149 and self=0 and r4wks01>0 then do;
  r4wks01=-3; r4ewks01=-3; end;
end;
end;

*** Include valid skips;
do over r4wks;
  if e200=-5 then do;
    r4wks=-5; r4swks=-5; r4ewks=-5;
  end;
end;

/* Check that no start date (except in ignore cases) comes
before the R's dli. */
do over starw;
  if r2int=-5 and r3int=-5 and intwk>0 and starw>0 and
    starw<r1int and ignore ne 1 then why1=1;
  if r2int>0 and r3int=-5 and intwk>0 and starw>0 and
    starw<r2int and ignore ne 1 then why2=1;
  if r3int>0 and intwk>0 and starw>0 and starw<r3int
    and ignore ne 1 then why3=1;
end;

/* Check that -3's in c.v. come from right places. The
cases below should all have missing years. */
array smofl (i) smofl1-smofl9;
array emofl (i) emofl1-emofl9;

```

```

do over starw;
if (r4wks98=-3 or r4wks99=-3 or r4wks00=-3 or
    r4wks01=-3) then do;
  if smofl=0 and emofl=0 and bgfl1_1=0 & bgfl1_2=0
    & bgfl1_3=0 & bgfl1_4=0 & bgfl1_5=0 &
    bgfl1_6=0 & bgfl1_7=0 & bgfl1_8=0 & bgfl1_9=0
    & bgfl1_10=0 & bgfl2_1=0 & bgfl2_2=0 &
    bgfl2_3=0 & bgfl2_4=0 & bgfl2_5=0 & bgfl3_1=0
    & bgfl3_2=0 & bgfl3_3=0 & bgfl4_1=0 &
    bgfl4_2=0 & bgfl4_3=0 & bgfl5_1=0 & bgfl6_1=0
    & egfl1_1=0 & egfl1_2=0 & egfl1_3=0 &
    egfl1_4=0 & egfl1_5=0 & egfl1_6=0 & egfl1_7=0
    & egfl1_8=0 & egfl1_9=0 & egfl1_10=0 &
    egfl2_1=0 & egfl2_2=0 & egfl2_3=0 & egfl2_4=0
    & egfl2_5=0 & egfl3_1=0 & egfl3_2=0 &
    egfl3_3=0 & egfl4_1=0 & egfl4_2=0 & egfl4_3=0
    & egfl5_1=0 & egfl6_1=0
  then yearmiss=1;
end;
end;

/* Round 3 variable */
array r3wks r3wks80-r3wks99 r3wks00;
/* Total created variable */
array twks twks80-twks99 twks00;

do over twks; r3wks=twks; end;

/* Define the following c.v.'s to match up the array sizes,
they are zero by definition.*/
r1wks99=0; r1wks00=0; r1wks01=0;
r2wks00=0; r2wks01=0;
r3wks01=0;

/* Round 1 created variable */
array r1wks (i) r1wks80-r1wks99 r1wks00 r1wks01;
/* Round 2 created variable */
array r2wks (i) r2wks80-r2wks99 r2wks00 r2wks01;
/* Round 3 created variable */
array r3wks (i) r3wks80-r3wks99 r3wks00 r3wks01;
/* Round 4 created variable, all jobs */
array r4wks (i) r4wks80-r4wks99 r4wks00 r4wks01;
/* Round 4 created variable, self-employed jobs */
array r4swks (i) r4swks80-r4swks99 r4swks00 r4swks01;
/* Round 4 created variable, employee jobs */
array r4ewks (i) r4ewks80-r4ewks99 r4ewks00 r4ewks01;
/* Total created variable, all jobs */
array twks (i) twks80-twks99 twks00 twks01;
/* Total created variable, self-employed jobs */
array tswks (i) tswks80-tswks99 tswks00 tswks01;
/* Total created variable, employee jobs */
array tewks (i) tewks80-tewks99 tewks00 tewks01;

/* Begin by splitting up periods where Round4 the dli c.v.
exclusively collect weeks worked information. Define the

```

dli interview year as the split. Any weeks worked information collected before the dli interview week should be independent of information collected in Round4. */

/* For the new Round 4 c.v.'s, we are assuming that all dli jobs are employee-type jobs. Thus, dli information will be added to the "all jobs" and "employee jobs" c.v.'s but not "self-employed jobs" c.v.'s. */

do over r4wks;
 twks=0; tswks=0; tewks=0;
 if r4wks>0 then do; twks=r4wks; end;
 if r4swks>0 then do; tswks=r4swks; end;
 if r4ewks>0 then do; tewks=r4ewks; end;
 if r3wks>0 and dliwk=r3int then do; twks=r3wks;
 tewks=r3wks; end;
 if r2wks>0 and dliwk=r2int then do; twks=r2wks;
 tewks=r2wks; end;
 if r1wks>0 and dliwk=r1int then do; twks=r1wks;
 tewks=r1wks; end;
 if r3wks>0 and r4wks>0 and dliwk=r3int then do;
 twks=r3wks+r4wks; end;
 if r3wks>0 and r4ewks>0 and dliwk=r3int then do;
 twks=r3wks+r4ewks; end;
 if r2wks>0 and r4wks>0 and dliwk=r2int then do;
 twks=r2wks+r4wks; end;
 if r2wks>0 and r4ewks>0 and dliwk=r2int then do;
 twks=r2wks+r4ewks; end;
 if r1wks>0 and r4wks>0 and dliwk=r1int then do;
 twks=r1wks+r4wks; end;
 if r1wks>0 and r4ewks>0 and dliwk=r1int then do;
 twks=r1wks+r4ewks; end;
end;

/* If either created variable from Round 4 or dli is a -2 or -3, then the overall created variable will be -2 or -3. */
do over r4wks;
 if r4wks in (-1,-2,-3) then do; twks=-3; end;
 if r4swks in (-1,-2,-3) then do; tswks=-3; end;
 if r4ewks in (-1,-2,-3) then do; tewks=-3; end;
 if r3wks in (-1,-2,-3) and dliwk=r3int then do; twks=-3;
 tewks=-3; end;
 if r2wks in (-1,-2,-3) and dliwk=r2int then do; twks=-3;
 tewks=-3; end;
 if r1wks in (-1,-2,-3) and dliwk=r1int then do; twks=-3;
 tewks=-3; end;
end;

/* Round 4 non-interview cases */
do over twks;
 if intwk=-5 then do; twks=-5; tswks=-5; tewks=-5; end;
end;

endsas;

NUMBER OF WEEKS WORKED SINCE LAST INTERVIEW

Variables Created: CV_WKSWK_DLI_ALL
CV_WKSWK_DLI_ET
CV_WKSWK_DLI_SE

Programs Used

This program uses emp_begin.sas as input (see the first page of this appendix for details).

For each individual, this program counts the number of weeks the respondent worked since the last interview. Three variables are created: one for all jobs, one for employee-type jobs, and one for self-employed jobs. **See the introduction to this appendix for important information about the universes for each of these variables.** Respondents not working in a given year are given a default value of zero.

<pre> array job1wks (i) wk1_1-wk1_1148; array job2wks (i) wk2_1-wk2_1148; array job3wks (i) wk3_1-wk3_1148; array job4wks (i) wk4_1-wk4_1148; array job5wks (i) wk5_1-wk5_1148; array job6wks (i) wk6_1-wk6_1148; array job7wks (i) wk7_1-wk7_1148; array job8wks (i) wk8_1-wk8_1148; array job9wks (i) wk9_1-wk9_1148; array alljobs (i) wks1-wks1148; array selfjobs (i) swks1-swks1148; array empjobs (i) ewks1-ewks1148; array starw (i) starw1-starw9; array stopw (i) stopw1-stopw9; array self (i) self1-self9; /* Create separate arrays for self employed jobs and employee jobs. The arrays are laid down in the same manner as "alljobs," with the self-employed dummy deciding which jobs are laid in each array. */ do over alljobs; alljobs=0; selfjobs=0; empjobs=0; end; /** Overlay multiple jobs over JOB 1 work weeks **/ do over job1wks; alljobs=job1wks; if self1=1 then do; selfjobs=job1wks; end; if self1=0 then do; empjobs=job1wks; end; end; do over alljobs; if job2wks=1 then do; alljobs=job2wks; end; if job2wks=1 & self2=1 then do; selfjobs=job2wks; end; if job2wks=1 & self2=0 then do; empjobs=job2wks; end; if job2wks=-3 & alljobs=0 then do; alljobs=job2wks; end; if job2wks=-3 & selfjobs=0 and self2=1 then do; selfjobs=job2wks; end; if job2wks=-3 & empjobs=0 and self2=0 then do; empjobs=job2wks; end; end; </pre>	<pre> /*At this point the program loops through the job2 code for jobs 3-9. Contact NLS User Services for details.*/ /** Calculate cumulative weeks on all jobs since dli ***/ allwks=0; selfwks=0; empwks=0; if dliwk>0 then do; if dliwk=>age14wk then do; do i=dliwk to 1148; if alljobs=1 then do; allwks=allwks+1; end; if selfjobs=1 then do; selfwks=selfwks+1; end; if empjobs=1 then do; empwks=empwks+1; end; end; do i=dliwk to 1148; if alljobs=-3 then do; allwks=-3; end; if selfjobs=-3 then do; selfwks=-3; end; if empjobs=-3 then do; empwks=-3; end; end; end; if age14wk>dliwk then do; do i=age14wk to 1148; if alljobs=1 then do; allwks=allwks+1; end; if selfjobs=1 then do; selfwks=selfwks+1; end; if empjobs=1 then do; empwks=empwks+1; end; end; do i=age14wk to 1148; if alljobs=-3 then do; allwks=-3; end; if selfjobs=-3 then do; selfwks=-3; end; if empjobs=-3 then do; empwks=-3; end; end; end; if starfl_1=1 then do; allwks=-3; end; /*repeat for starfl_2 through starfl_9*/ if stopfl_1=1 then do; allwks=-3; end; /*repeat for stopfl_2 through stopfl_9*/ if e200=-5 then do; allwks=-5; selfwks=-5; empwks=-5; end; endsas; </pre>
---	---

NUMBER OF WEEKS WORKED AGE 14–19, WEEKS WORKED AT ALL JOBS SINCE AGE 20, AND WEEKS WORKED AT EMPLOYEE-TYPE JOBS SINCE AGE 20

Variables Created: CV_WKSWK_TEEN
CV_WKSWK_ADULT_ET
CV_WKSWK_ADULT_ALL

Programs Used

This program uses **emp_begin.sas** as input (see the first page of this appendix for details).

For each respondent, this program creates a variable counting the number of weeks worked at any employee-type job between the ages of 14 and 19. It then creates two variables counting weeks worked from age 20 to the present; one totals all jobs and one presents only employee jobs. See the **introduction to this appendix for important information about the universes for each of these variables**. Respondents not working are given a default value of zero; respondents who have not yet reached age 20 receive a valid skip (-4) on the latter two variables.

<pre> array job1wks (i) wk1_1-wk1_1148; array job2wks (i) wk2_1-wk2_1148; array job3wks (i) wk3_1-wk3_1148; array job4wks (i) wk4_1-wk4_1148; array job5wks (i) wk5_1-wk5_1148; array job6wks (i) wk6_1-wk6_1148; array job7wks (i) wk7_1-wk7_1148; array job8wks (i) wk8_1-wk8_1148; array job9wks (i) wk9_1-wk9_1148; array alljobs (i) wks1-wks1148; array empjobs (i) ewks1-ewks1148; array starw (i) starw1-starw9; array stopw (i) stopw1-stopw9; array starfl (i) starfl_1-starfl_9; array stopfl (i) stopfl_1-stopfl_9; /** Overlay multiple jobs over JOB 1 work weeks **/ do over empjobs; empjobs=0; end; do over job1wks; if self1=0 then do; empjobs=job1wks; end; end; do over empjobs; if job2wks=1 and self2=0 then do; empjobs=job2wks; end; if job2wks=-3 and self2=0 and empjobs=0 then do; empjobs=job2wks; end; end; /*Repeat the job 2 code above for jobs 3-9; contact NLS User Services for more information.*/ do over alljobs; alljobs=0; end; do over job1wks; alljobs=job1wks; end; do over alljobs; if job2wks=1 then do;</pre>	<pre> alljobs=job2wks; end; if job2wks=-3 and alljobs=0 then do; alljobs=job2wks; end; end; /*Repeat the job 2 code above for jobs 3-9; contact NLS User Services for more information.*/ /*Calculate cumulative weeks on all jobs between age 14 and age 19, and on all jobs and employee jobs since age 20*/ /* r4emp14 is the variable that counts the number of weeks worked since respondent's 14th birthday. This is changed from wks14 to distinguish it from a member in the alljobs array. */ r4emp14=0; r4wks20=0; r4emp20=0; /* Note the max for age20wk is 1305. */ if age14wk>0 and (age20wk-1)<=1148 then do; do i=age14wk to (age20wk-1); if empjobs=1 then do; r4emp14=r4emp14+1; end; if empjobs=-3 then do; r4emp14=-3; end; end; end; if age14wk>0 and (age20wk-1)>1148 then do; do i=age14wk to 1148; if empjobs=1 then do; r4emp14=r4emp14+1; end; if empjobs=-3 then do; r4emp14=-3; end; end; end; if age14wk>0 and age20wk<=1148 then do; do i=age20wk to 1148; if alljobs=1 then do; r4wks20=r4wks20+1; end; if alljobs=-3 then do; r4wks20=-3; end; end; end;</pre>
---	---

```

end;
end;

if age14wk>0 and age20wk<=1148 then do;
  do i=age20wk to 1148;
    if empjobs=1 then do; r4emp20=r4emp20+1; end;
    if empjobs=-3 then do; r4emp20=-3; end;
  end;
end;

/* lowest age14wk is 731, lowest age20wk is 1044*/
do i=1 to 9;
if starfl=1 and r4emp14>0 then do;
  if age14wk<784 then do;
    if stopw>730 then do; r4emp14=-3; end;
  end;
  if age14wk<836 then do;
    if stopw>783 then do; r4emp14=-3; end;
  end;
  if age14wk<888 then do;
    if stopw>835 then do; r4emp14=-3; end;
  end;
  if age14wk<940 then do;
    if stopw>887 then do; r4emp14=-3; end;
  end;
  if age14wk<991 then do;
    if stopw>939 then do; r4emp14=-3; end;
  end;
  if age14wk<1044 then do;
    if stopw>990 then do; r4emp14=-3; end;
  end;
  if age14wk<1097 then do;
    if stopw>1043 then do; r4emp14=-3; end;
  end;
  if age14wk<1149 then do;
    if stopw>1096 then do; r4emp14=-3; end;
  end;
end;

if starfl=1 and r4wks20>0 then do;
  if age20wk<1097 then do;
    if starw<1097 then do; r4wks20=-3; r4emp20=-3; end;
  end;
  if age20wk<1149 then do;
    if starw<1149 then do; r4wks20=-3; r4emp20=-3; end;
  end;
end;

*** Include noninterviews;
if e200=-5 then do;
  r4emp14=-5; r4wks20=-5; r4emp20=-3;
end;

/* This program calculates the total number of weeks
worked since age 14. Since the tenure created variables
from Round 4 and dli Round are only calculated during
the interview period, this program sums up these tenure
dates to get a total weeks count. */

r3wks14=totwks14;
r3wks20=totwks20;
drop totwks14 totwks20;

/* Initialize created variable for both rounds to zero.*/
temp14=0;
twks20=0;
temp20=0;

/* By the construction of the Round 4 created variable, we
can simply add the two created variables from Round 4
and dli Round if they are both positive. If one is positive
and one is zero, then the positive value will be the total
created variable for both rounds. If neither is positive,
then the total created variable for both rounds will be zero.
If either the Round 4 c.v. or the dli c.v. is invalid, then the
total c.v. will be -3. */

```

Appendix 2: Employment Variable Creation

```
if r4emp14=>0 then do; temp14=r4emp14; end;
if r4emp14=-3 then do; temp14=-3; end;

/* dli is Round 3 */
if r3wks14=>0 and r4emp14=>0 then do;
  temp14=r3wks14+r4emp14;
end;
if r3wks14=-3 then do; temp14=-3; end;

/* dli is Round 2 */
if r2wks14=>0 and r4emp14=>0 and r3int=-5 then do;
  temp14=r2wks14+r4emp14;
end;
if r2wks14=-3 and r3int=-5 then do; temp14=-3; end;

/* dli is Round 1 */
if r1wks14=>0 and r4emp14=>0 and r3int=-5 and
r2int=-5 then do;
  temp14=r1wks14+r4emp14;
end;
if r1wks14=-3 and r3int=-5 and r2int=-5 then do;
  temp14=-3;
end;

if r4wks20>0 then twks20=r4wks20;
if r3wks20>0 then twks20=r3wks20;
if r4wks20>0 and r3wks20>0 then do;
  twks20=r4emp20+r3wks20;
end;
if r4emp20=-3 or r3wks20=-3 then do;
  twks20=-3;
end;

if r4emp20>0 then temp20=r4emp20;
if r3wks20>0 then temp20=r3wks20;
if r4emp20>0 and r3wks20>0 then do;
  temp20=r4emp20+r3wks20;
end;
if r4emp20=-3 or r3wks20=-3 then do;
  temp20=-3;
end;

if uid1=-5 then do;
  temp14=-5; twks20=-5; temp20=-5;
end;

/* For respondents younger than 20 at interview date,
their thrs14 c.v. will be -4. */
if age20wk>intwk and intwk ne -5 then do;
  twks20=-4; temp20=-4;
end;

endsas;
```

WEEKS WORKED AT EMPLOYEE JOB #X DURING 19XX/20XX

Variables Created: CV_WKSWK_JOB_YR.01.80 – CV_WKSWK_JOB_YR.01.01
CV_WKSWK_JOB_YR.02.80 – CV_WKSWK_JOB_YR.02.01 etc. through job #9

Programs Used

This program uses **emp_begin.sas** as input (see the first page of this appendix for details).

This program creates variables for each of the respondent's jobs counting the number of weeks worked in each calendar year. A variable is created for each potential job even if the respondent has worked no jobs in a given year with the default value set to zero (0). The most jobs held by any respondent in round 4 was nine, so variables are created for nine jobs for each respondent.

```
/* Section 1: Create the Round3 variable for number of
weeks worked at a given job in a given year. */

array starw (i) starw1-starw9;
array stopw (i) stopw1-stopw9;
array job1wks (i) wk1_1-wk1_1148;
/*repeat for jobs 2-9 (e.g. job2wks = wk2_1-wk2_1148)*/
array starfl (i) starfl_1-starfl_9;
array stopfl (i) stopfl_1-stopfl_9;
array r4wks80 (i) r4wks801-r4wks809;
array r4wks81 (i) r4wks811-r4wks819;
/*and so on through*/
array r4wks01 (i) r4wks011-r4wks019;

/** Calculate cumulative weeks on individual jobs for
each year **/ 

/* 1980 */

r4wks801=0; r4wks802=0; r4wks803=0; r4wks804=0;
r4wks805=0; r4wks806=0; r4wks807=0; r4wks808=0;
r4wks809=0;

do i=1 to 52;
  if job1wks=1 then do; r4wks801=r4wks801+1; end;
  if job2wks=1 then do; r4wks802=r4wks802+1; end;
  if job3wks=1 then do; r4wks803=r4wks803+1; end;
  if job4wks=1 then do; r4wks804=r4wks804+1; end;
  if job5wks=1 then do; r4wks805=r4wks805+1; end;
  if job6wks=1 then do; r4wks806=r4wks806+1; end;
  if job7wks=1 then do; r4wks807=r4wks807+1; end;
  if job8wks=1 then do; r4wks808=r4wks808+1; end;
  if job9wks=1 then do; r4wks809=r4wks809+1; end;

  if job1wks=-3 then do; r4wks801=-3; end;
  if job2wks=-3 then do; r4wks802=-3; end;
  if job3wks=-3 then do; r4wks803=-3; end;
  if job4wks=-3 then do; r4wks804=-3; end;
  if job5wks=-3 then do; r4wks805=-3; end;
  if job6wks=-3 then do; r4wks806=-3; end;
  if job7wks=-3 then do; r4wks807=-3; end;
  if job8wks=-3 then do; r4wks808=-3; end;
  if job9wks=-3 then do; r4wks809=-3; end;
```

```
end;

/*At this point the program repeats the above code for
each year through 2001. Due to space considerations this
code is not included here. Researchers needing the
complete code should contact NLS User Services. The
variables and "do i" statements for each year are as follows:

1981   wksw811-wksw819    do i=53 to 104
1982   wksw821-wksw829    do i=105 to 156
1983   wksw831-wksw839    do i=157 to 209
1984   wksw841-wksw849    do i=210 to 261
1985   wksw851-wksw859    do i=262 to 313
1986   wksw861-wksw869    do i=314 to 365
1987   wksw871-wksw879    do i=366 to 417
1988   wksw881-wksw889    do i=418 to 470
1989   wksw891-wksw899    do i=471 to 522
1990   wksw901-wksw909    do i=523 to 574
1991   wksw911-wksw919    do i=575 to 626
1992   wksw921-wksw929    do i=627 to 678
1993   wksw931-wksw939    do i=679 to 730
1994   wksw941-wksw949    do i=731 to 783
1995   wksw951-wksw959    do i=784 to 835
1996   wksw961-wksw969    do i=836 to 887
1997   wksw971-wksw979    do i=888 to 939
1998   wksw981-wksw989    do i=940 to 991
1999   wksw991-wksw999    do i=992 to 1044
2000   wksw001-wksw009    do i=1045 to 1098  */
2000   wksw011-wksw019    do i=1097 to 1148  */

do i=1 to 9;
/*start date invalid*/
if starfl=1 then do;
  if r4wks80>0 & stopw>1 then do; r4wks80=-3; end;
  if r4wks81>0 & stopw>52 then do; r4wks81=-3; end;
  if r4wks82>0 & stopw>104 then do; r4wks82=-3; end;
  if r4wks83>0 & stopw>156 then do; r4wks83=-3; end;
  if r4wks84>0 & stopw>209 then do; r4wks84=-3; end;
  if r4wks85>0 & stopw>261 then do; r4wks85=-3; end;
  if r4wks86>0 & stopw>313 then do; r4wks86=-3; end;
  if r4wks87>0 & stopw>365 then do; r4wks87=-3; end;
  if r4wks88>0 & stopw>417 then do; r4wks88=-3; end;
  if r4wks89>0 & stopw>470 then do; r4wks89=-3; end;
  if r4wks90>0 & stopw>522 then do; r4wks90=-3; end;
  if r4wks91>0 & stopw>574 then do; r4wks91=-3; end;
```

```

if r4wks92>0 & stopw>626 then do; r4wks92=-3; end;
if r4wks93>0 & stopw>678 then do; r4wks93=-3; end;
if r4wks94>0 & stopw>730 then do; r4wks94=-3; end;
if r4wks95>0 & stopw>783 then do; r4wks95=-3; end;
if r4wks96>0 & stopw>835 then do; r4wks96=-3; end;
if r4wks97>0 & stopw>887 then do; r4wks97=-3; end;
if r4wks98>0 & stopw>939 then do; r4wks98=-3; end;
if r4wks99>0 & stopw>991 then do; r4wks99=-3; end;
if r4wks00>0 & stopw>1043 then do; r4wks00=-3; end;
if r4wks01>0 & stopw>1096 then do; r4wks01=-3; end;
end;

/* stop date invalid */
if stopfl=1 then do;
  if r4wks80>0 & starw<53 then do; r4wks80=-3; end;
  if r4wks81>0 & starw<105 then do; r4wks81=-3; end;
  if r4wks82>0 & starw<157 then do; r4wks82=-3; end;
  if r4wks83>0 & starw<210 then do; r4wks83=-3; end;
  if r4wks84>0 & starw<262 then do; r4wks84=-3; end;
  if r4wks85>0 & starw<314 then do; r4wks85=-3; end;
  if r4wks86>0 & starw<366 then do; r4wks86=-3; end;
  if r4wks87>0 & starw<418 then do; r4wks87=-3; end;
  if r4wks88>0 & starw<471 then do; r4wks88=-3; end;
  if r4wks89>0 & starw<523 then do; r4wks89=-3; end;
  if r4wks90>0 & starw<575 then do; r4wks90=-3; end;
  if r4wks91>0 & starw<627 then do; r4wks91=-3; end;
  if r4wks92>0 & starw<679 then do; r4wks92=-3; end;
  if r4wks93>0 & starw<731 then do; r4wks93=-3; end;
  if r4wks94>0 & starw<784 then do; r4wks94=-3; end;
  if r4wks95>0 & starw<836 then do; r4wks95=-3; end;
  if r4wks96>0 & starw<888 then do; r4wks96=-3; end;
  if r4wks97>0 & starw<940 then do; r4wks97=-3; end;
  if r4wks98>0 & starw<991 then do; r4wks98=-3; end;
  if r4wks99>0 & starw<1044 then do; r4wks99=-3; end;
  if r4wks00>0 & starw<1097 then do; r4wks00=-3; end;
  if r4wks01>0 & starw<1149 then do; r4wks01=-3; end;
end;
end;

if e200=-5 then do;
  r4wks801=-5; r4wks802=-5; r4wks803=-5; r4wks804=-5;
  r4wks805=-5; r4wks806=-5; r4wks807=-5;
  r4wks808=-5; r4wks809=-5;
/* and so on through*/
  r4wks011=-5; r4wks012=-5; r4wks013=-5; r4wks014=-5;
  r4wks015=-5; r4wks016=-5; r4wks017=-5;
  r4wks018=-5; r4wks019=-5;
end;

/*Now merge in the information from previous rounds*/
/* Round 3 variable */
array r3wks80 r3wks801-r3wks809;
array r3wks81 r3wks811-r3wks819;
array r3wks82 r3wks821-r3wks829;
array r3wks83 r3wks831-r3wks839;
array r3wks84 r3wks841-r3wks849;
array r3wks85 r3wks851-r3wks859;

```

```

array r3wks86 r3wks861-r3wks869;
array r3wks87 r3wks871-r3wks879;
array r3wks88 r3wks881-r3wks889;
array r3wks89 r3wks891-r3wks899;
array r3wks90 r3wks901-r3wks909;
array r3wks91 r3wks911-r3wks919;
array r3wks92 r3wks921-r3wks929;
array r3wks93 r3wks931-r3wks939;
array r3wks94 r3wks941-r3wks949;
array r3wks95 r3wks951-r3wks959;
array r3wks96 r3wks961-r3wks969;
array r3wks97 r3wks971-r3wks979;
array r3wks98 r3wks981-r3wks989;
array r3wks99 r3wks991-r3wks999;
array r3wks00 r3wks001-r3wks009;

/* Total created variable */
array twks80 twks801-twks809;
array twks81 twks811-twks819;
array twks82 twks821-twks829;
array twks83 twks831-twks839;
array twks84 twks841-twks849;
array twks85 twks851-twks859;
array twks86 twks861-twks869;
array twks87 twks871-twks879;
array twks88 twks881-twks889;
array twks89 twks891-twks899;
array twks90 twks901-twks909;
array twks91 twks911-twks919;
array twks92 twks921-twks929;
array twks93 twks931-twks939;
array twks94 twks941-twks949;
array twks95 twks951-twks959;
array twks96 twks961-twks969;
array twks97 twks971-twks979;
array twks98 twks981-twks989;
array twks99 twks991-twks999;
array twks00 twks001-twks009;

do over twks80;
  r3wks80=twks80;           r3wks81=twks81;
  r3wks82=twks82;           r3wks83=twks83;
  r3wks84=twks84;           r3wks85=twks85;
  r3wks86=twks86;           r3wks87=twks87;
  r3wks88=twks88;           r3wks89=twks89;
  r3wks90=twks90;           r3wks91=twks91;
  r3wks92=twks92;           r3wks93=twks93;
  r3wks94=twks94;           r3wks95=twks95;
  r3wks96=twks96;           r3wks97=twks97;
  r3wks98=twks98;           r3wks99=twks99;
  r3wks00=twks00;
end;
/* WKS945 is the Round 2 created variable for 5th job
(i.e. 5th UID code) in 1994. WKS945 is the Round 3
created variable of the same description. r1wks594 is the
variable that combines these two created variables (of the
same description). These infile variable names are
```

Appendix 2: Employment Variable Creation

confusing, so rename them to more familiar "r1wks, r2wks" etc. r1wks remains the total created variable. */

/* Note there are 7 jobs in Round 1, array matched to make programming easier */

```
array r1uid (i) r1uid1-r1uid9;
array r2uid (i) r2uid1-r2uid9;
array r3uid (i) r3uid1-r3uid9;
array r4uid (i) uid1-uid9;
```

/* Arrange the jobs by year for Round 1*/

```
array r1wks80 (i) r1wks801-r1wks809;
array r1wks81 (i) r1wks811-r1wks819;
/*and so on through*/
array r1wks99 (i) r1wks991-r1wks999;
```

/* Arrange the jobs by year for Round 2*/

```
array r2wks80 (i) r2wks801-r2wks809;
array r2wks81 (i) r2wks811-r2wks819;
/*and so on through*/
array r2wks00 (i) r2wks001-r2wks009;
```

/* Arrange the jobs by year for Round 3*/

```
array r3wks80 (i) r3wks801-r3wks809;
array r3wks81 (i) r3wks811-r3wks819;
/*and so on through*/
array r3wks00 (i) r3wks001-r3wks009;
```

/* Arrange the jobs by year for Round 4*/

```
array r4wks80 (i) r4wks801-r4wks809;
/*and so on through*/
array r4wks01 (i) r4wks011-r4wks019;
```

/* Arrange the jobs by year for total c.v. */

```
array twks80 (i) twks801-twks809;
array twks81 (i) twks811-twks819;
/*and so on through*/
array twks01 (i) twks011-twks019;
```

/* Initialize total weeks count to zero */

```
do i=1 to 9;
twks80=0; twks81=0; twks82=0; twks83=0; twks84=0;
twks85=0; twks86=0; twks87=0; twks88=0;
twks89=0; twks90=0; twks91=0; twks92=0;
twks93=0; twks94=0; twks95=0; twks96=0;
twks97=0; twks98=0; twks99=0; twks00=0;
twks01=0;
end;
```

/* Accounts for Round 3 non-interview cases */

```
do i=1 to 9;
if e200=-5 then do;
twks80=-5; twks81=-5; twks82=-5; twks83=-5;
twks84=-5; twks85=-5; twks86=-5; twks87=-5;
twks88=-5; twks89=-5; twks90=-5; twks91=-5;
twks92=-5; twks93=-5; twks94=-5; twks95=-5;
```

```
twks96=-5; twks97=-5; twks98=-5; twks99=-5;
twks00=-5; twks01=-5;
```

```
end;
end;
```

/* Begin by updating the total created variable if the Round 4 created variable is positive. For this variable, we are only worried about Round 4 jobs and previous round jobs that were worked in Round 4. If a invalid negative value is given (don't know, refusal, or non-interview) then total created variable will be that negative value. */

/* 1980 */

```
do i=1 to 9;
if r4wks80=>0 then do; twks80=r4wks80; end;
if -4<r4wks80<0 then do; twks80=-3; end;
end;
```

/* 1981 */

```
do i=1 to 9;
if r4wks81=>0 then do; twks81=r4wks81; end;
if -4<r4wks81<0 then do; twks81=-3; end;
end;
```

/*and so on through*/

```
/* 1996 */
do i=1 to 9;
if r4wks96=>0 then do; twks96=r4wks96; end;
if -4<r4wks96<0 then do; twks96=-3; end;
end;
```

/* Since 1997-2000 are the interview years, we need to account for both Round 4 and dli created variables. Begin with negative conditions for the R4 c.v.*/

/* 1997 */

```
do i=1 to 9;
if -4<r4wks97<0 then do; twks97=-3; end;
if r4wks97=>0 then do; twks97=r4wks97; end;
end;
```

/* 1998 */

```
do i=1 to 9;
if -4<r4wks98<0 then do; twks98=-3; end;
if r4wks98=>0 then do; twks98=r4wks98; end;
end;
```

/* 1999 */

```
do i=1 to 9;
if -4<r4wks99<0 then do; twks99=-3; end;
if r4wks99=>0 then do; twks99=r4wks99; end;
end;
```

/* 2000 */

```
do i=1 to 9;
if -4<r4wks00<0 then do; twks00=-3; end;
```

```

if r4wks00>0 then do; twks00=r4wks00; end;
end;

/* 2001 */
do i=1 to 9;
  if -4<r4wks01<0 then do; twks01=-3; end;
  if r4wks01>0 then do; twks01=r4wks01; end;
end;

array match1 (i) match11-match19;
array match2 (i) match21-match29;
  /*and so on through*/
array match9 (i) match91-match99;

do i=1 to 9;
  match1=0; match2=0; match3=0; match4=0;
  match5=0; match6=0; match7=0; match8=0;
  match9=0;
end;

/* Define match14 as the dummy variable that equals one
   when the first job on the R1 UID roster and the
   fourth job on the R2 roster have the same UID. */

/* If any UID from the first position in a previous round
   matches with any UID in Round 4. Need to be
   sure to find the most recent round that matches. */
do over r4uid;
  if r3uid1>0 and r4uid>0 and r3int>0 and intwk>0 then
    do;
      if r3uid1=r4uid then do; match1=3; end;
    end;
  if r2uid1>0 and r4uid>0 and match1=0 and intwk>0
    then do;
      if r2uid1=r4uid then do; match1=2; end;
    end;
  if r1uid1>0 and r4uid>0 and match1=0 and intwk>0
    then do;
      if r1uid1=r4uid then do; match1=1; end;
    end;
end;

/* Match UID from the second position in previous
   rounds to UIDs in Round 4 */
do over r4uid;
  if r3uid2>0 and r4uid>0 and r3int>0 and intwk>0 then
    do;
      if r3uid2=r4uid then do; match2=3; end;
    end;
  if r2uid2>0 and r4uid>0 and match2=0 and intwk>0
    then do;
      if r2uid2=r4uid then do; match2=2; end;
    end;
  if r1uid2>0 and r4uid>0 and match2=0 and intwk>0
    then do;
      if r1uid2=r4uid then do; match2=1; end;
    end;
end;

end;
/*repeat the above code for UIDs in positions 3-9 in
previous rounds*/

r4match_1=max(match11,match21,match31,match41,
  match51,match61,match71,match81,match91);
r4match_2=max(match12,match22,match32,match42,
  match52,match62,match72,match82,match92);
  /* and so on through */
r4match_9=max(match19,match29,match39,match49,
  match59,match69,match79,match89,match99);

if match11<r4match_1 then match11=0;
if match21<r4match_1 then match21=0;
  /* and so on through */
if match91<r4match_1 then match91=0;

if match12<r4match_2 then match12=0;
if match22<r4match_2 then match22=0;
  /* and so on through */
if match92<r4match_2 then match92=0;

if match13<r4match_3 then match13=0;
if match23<r4match_3 then match23=0;
  /* and so on through */
if match93<r4match_3 then match93=0;

if match14<r4match_4 then match14=0;
if match24<r4match_4 then match24=0;
  /* and so on through */
if match94<r4match_4 then match94=0;

if match15<r4match_5 then match15=0;
if match25<r4match_5 then match25=0;
  /* and so on through */
if match95<r4match_5 then match95=0;

if match16<r4match_6 then match16=0;
if match26<r4match_6 then match26=0;
  /* and so on through */
if match96<r4match_6 then match96=0;

if match17<r4match_7 then match17=0;
if match27<r4match_7 then match27=0;
  /* and so on through */

if match18<r4match_8 then match18=0;
if match28<r4match_8 then match28=0;
  /* and so on through */
if match98<r4match_8 then match98=0;

if match19<r4match_9 then match19=0;
if match29<r4match_9 then match29=0;
  /* and so on through */
if match99<r4match_9 then match99=0;

```

```

/* Now reassign total created variable if there is an
   identical job was worked in both rounds. Using the
   match variable, a job worked in both rounds will
   update the total created variable. Need to add
   positivity condition on the weeks worked values
   since some of the values equal -3. */

/* For respondents last interviewed in Round 3, 2000 */
if r3int_y=2000 then do;
do over match1;
  if match1=3 then do;
    if r3wks801 not in (-5,-4,0..) then
      twks80=r3wks801;
    if r3wks811 not in (-5,-4,0..) then
      twks81=r3wks811;
    /*and so on through*/
    if r3wks991 not in (-5,-4,0..) then
      twks99=r3wks991;
  end;
end;

do over match1;
  if match1=3 and r4wks00=>0 and r3wks001=>0 then
    do; twks00=r3wks001+r4wks00; end;
  if match1=3 and (-4<r4wks00<0 or -4<r3wks001<0)
    then do; twks00=-3; end;
end;

/*At this point the program repeats the above code for
match2-match9. Contact NLS User Services for details.*/
end;

/* For respondents last interviewed in Round 3, 1999 */
if r3int_y=1999 then do;
do over match1;
  if match1=3 then do;
    if r3wks801 not in (-5,-4,0..) then
      twks80=r3wks801;
    if r3wks811 not in (-5,-4,0..) then
      twks81=r3wks811;
    /*and so on through*/
    if r3wks981 not in (-5,-4,0..) then
      twks98=r3wks981;
  end;
end;

do over match1;
  if match1=3 and r4wks99=>0 and r3wks991=>0 then
    do; twks99=r3wks991+r4wks99; end;
  if match1=3 and (-4<r4wks99<0 or -4<r3wks991<0)
    then do; twks99=-3; end;
end;

/*At this point the program repeats the above code for
match2-match9. Contact NLS User Services for details.*/

```

```

end;

/* For respondents last interviewed in Round 2, 1998 */
if (r2int_y=1998) then do;
do over match1;
  if match1=2 then do;
    if r2wks801 not in (-5,-4,0..) then
      twks80=r2wks801;
    if r2wks811 not in (-5,-4,0..) then
      twks81=r2wks811;
    /*and so on through*/
    if r2wks971 not in (-5,-4,0..) then
      twks97=r2wks971;
  end;
end;

do over match1;
  if match1=2 and r4wks98=>0 and r2wks981=>0 then
    do; twks98=r2wks981+r4wks98; end;
  if match1=2 and (-4<r4wks98<0 or -4<r2wks981<0)
    then do; twks98=-3; end;
end;

/*At this point the program repeats the above code for
match2-match9. Contact NLS User Services for details.*/
end;

/* For respondents last interviewed in Round 2, 1999 */
if r2int_y=1999 then do;
do over match1;
  if match1=2 then do;
    if r2wks801 not in (-5,-4,0..) then
      twks80=r2wks801;
    if r2wks811 not in (-5,-4,0..) then
      twks81=r2wks811;
    /*and so on through*/
    if r2wks981 not in (-5,-4,0..) then
      twks98=r2wks981;
  end;
end;

do over match1;
  if match1=2 and r4wks99=>0 and r2wks991=>0 then
    do; twks99=r2wks991+r4wks99; end;
  if match1=2 and (-4<r4wks99<0 or -4<r2wks991<0)
    then do; twks99=-3; end;
end;

/*At this point the program repeats the above code for
match2-match9. Contact NLS User Services for details.*/
end;

/* For respondents last interviewed in Round 1, 1997 */
if r1int_y=1997 and intwk>0 then do;
```

```

do over match1;
if match1=1 then do;
  if r1wks801 not in (-5,-4,0,.) then
    twks80=r1wks801;
  if r1wks811 not in (-5,-4,0,.) then
    twks81=r1wks811;
  /*and so on through*/
  if r1wks961 not in (-5,-4,0,.) then
    twks96=r1wks961;
end;
end;

do over match1;
  if match1=1 and r4wks97=>0 and r1wks971=>0 then
    do; twks97=r1wks971+r4wks97; end;
  if match1=1 and (-4<r4wks97<0 or -4<r1wks971<0)
    then do; twks97=-3; end;
end;

/*At this point the program repeats the above code for
match2-match9. Contact NLS User Services for details.*/
end;

/* For respondents last interviewed in Round 1, 1998 */
if r1int_y=1998 and intwk>0 then do;
  do over match1;
    if match1=1 then do;
      if r1wks801 not in (-5,-4,0,.) then
        twks80=r1wks801;
      if r1wks811 not in (-5,-4,0,.) then
        twks81=r1wks811;
      /*and so on through*/
      if r1wks971 not in (-5,-4,0,.) then
        twks97=r1wks971;
      end;
    end;
  do over match1;
    if match1=1 and r4wks98=>0 and r1wks981=>0 then
      do; twks98=r1wks981+r4wks98; end;
    if match1=1 and (-4<r4wks98<0 or -4<r1wks981<0)
      then do; twks98=-3; end;
  end;
  /*At this point the program repeats the above code for
  match2-match9. Contact NLS User Services for details.*/
end;
  /*flag problem cases where total weeks count exceeds 52*/
  flag97=0; flag98=0; flag99=0;
  do i=1 to 9;
    if twks97>52 then flag97=1;
    if twks98>52 then flag98=1;
    if twks99>52 then flag99=1;
    if twks00>53 then flag00=1;
  end;
  endsas;

```

TOTAL TENURE AT JOB #X AS OF THE SURVEY DATE

Variables Created: CV_WKSWK_JOB_DLI.01 – CV_WKSWK_JOB_DLI.09

Programs Used

This program uses `emp_begin.sas` as input (see the first page of this appendix for details).

This program creates a variable for each job calculating the total length of job tenure in weeks (excluding within-job gaps) since the respondent's 14th birthday. A variable is created for each potential job even if the respondent has no data for that job, with the default value set to zero (0). The most jobs held by any respondent as of round 2 was nine, so variables are created for nine jobs for each respondent.

```

/* Section 1: Create a variable for each potential job (9)
   relating the total length of job tenure in weeks excluding
   within-job gaps.*/

array starw (i) starw1-starw9;
array job1wks (i) wk1_1-wk1_1148;
array job2wks (i) wk2_1-wk2_1148;
/*and so on through*/
array job9wks (i) wk9_1-wk9_1148;
array starfl (i) starfl_1-starfl_9;
array stopfl (i) stopfl_1-stopfl_9;

/* Calculate cumulative weeks on individual jobs for all
   years */
r4ten1=0; r4ten2=0; r4ten3=0; r4ten4=0; r4ten5=0;
r4ten6=0; r4ten7=0; r4ten8=0; r4ten9=0;

do i=1 to 1148;
  if job1wks=1 then do; r4ten1=r4ten1+1; end;
  if job2wks=1 then do; r4ten2=r4ten2+1; end;
  /*and so on through*/
  if job9wks=1 then do; r4ten9=r4ten9+1; end;
end;

flag=0;

do i=1 to 1148;
  if job1wks=-3 then do; r4ten1=-3; flag=1; end;
  if job2wks=-3 then do; r4ten2=-3; flag=2; end;
  /*and so on through*/
  if job9wks=-3 then do; r4ten9=-3; flag=9; end;
end;

do i=1 to 1096;
  if starfl_1=1 then do; r3ten1=-3; end;
  if starfl_2=1 then do; r3ten2=-3; end;
  /*and so on through*/
  if starfl_9=1 then do; r3ten9=-3; end;
  if stopfl_1=1 then do; r3ten1=-3; end;
  if stopfl_2=1 then do; r3ten2=-3; end;
  /*and so on through*/
  if stopfl_9=1 then do; r3ten9=-3; end;
end;

```

```

if e200=-5 then do;
  r4ten1=-5; r4ten2=-5; r4ten3=-5; r4ten4=-5; r4ten5=-5;
  r4ten6=-5; r4ten7=-5; r4ten8=-5; r4ten9=-5;
end;

array r3ten (i) r3ten1-r3ten9;
array totten (i) totten1-totten9;
do over totten; r3ten=totten; end;

/* 2-26-01 Begin by renaming the infile variables */
array r1ten (i) r1ten1-r1ten9; /* New R1 c.v. name */
array r2ten (i) r2ten1-r2ten9; /* New R2 c.v. name */
array r3ten (i) r3ten1-r3ten9; /* New R3 c.v. name */
array r4ten (i) r4ten1-r4ten9; /* New R4 c.v. name */
array totten (i) totten1-totten9; /* Total c.v. name */

do i=1 to 9; totten=0; end;

array r1uid (i) r1uid1-r1uid7;
array r2uid (i) r2uid1-r2uid9;
array r3uid (i) r3uid1-r3uid9;
array r4uid (i) uid1-uid9;

array match1 (i) match11-match19;
array match2 (i) match21-match29;
/*and so on through*/
array match9 (i) match91-match99;

do i=1 to 9;
  match1=0; match2=0; match3=0; match4=0; match5=0;
  match6=0; match7=0; match8=0; match9=0;
end;

/* Define match14 as the dummy variable that equals one
   when the first job on the dli UID roster and the fourth
   job on the Round4 roster have the same UID. */

/* If any UID from the first position in a previous round
   matches with any UID in Round 4. Need to make sure
   we get the most recent round the job was reported in. */
do over r4uid;
  if r3uid1>0 and r4uid>0 and r3int>0 and intwk>0 then
    do;
      if r3uid1=r4uid then do; match1=3; end;

```

```

end;
if r2uid1>0 and r4uid>0 and match1=0 and intwk>0
    then do;
    if r2uid1=r4uid then do; match1=2; end;
end;
if r1uid1>0 and r4uid>0 and match1=0 and intwk>0
    then do;
    if r1uid1=r4uid then do; match1=1; end;
end;
end;

/* If any UID from the second position in a previous
   Round matches with any UID in Round 4 */
do over r4uid;
if r3uid2>0 and r4uid>0 and r3int>0 and intwk>0 then
    do;
    if r3uid2=r4uid then do; match2=3; end;
end;
if r2uid2>0 and r4uid>0 and match2=0 and intwk>0
    then do;
    if r2uid2=r4uid then do; match2=2; end;
end;
if r1uid2>0 and r4uid>0 and match2=0 and intwk>0
    then do;
    if r1uid2=r4uid then do; match2=1; end;
end;
end;

/*repeat the above code for UIDs in positions 3-9 in
   previous rounds*/

r4match_1=max(match11,match21,match31,match41,
    match51,match61,match71,match81,match91);
r4match_2=max(match12,match22,match32,match42,
    match52,match62,match72,match82,match92);
/*and so on through*/
r4match_9=max(match19,match29,match39,match49,
    match59,match69,match79,match89,match99);

if match11<r4match_1 then match11=0;
if match21<r4match_1 then match21=0;
/*and so on through*/
if match91<r4match_1 then match91=0;

if match12<r4match_2 then match12=0;
if match22<r4match_2 then match22=0;
/*and so on through*/
if match92<r4match_2 then match92=0;

/*repeat the above code for jobs 2-9*/

/* There can only be 9 possible jobs for the respondent,
since we are not counting jobs that were only worked in
previous rounds. */
do over totten; totten=r4ten; end;

/* Now reassign total tenure if there is a job was worked in
2+ rounds. Using the match variable, a job worked in 2+
round 4 & the dli round will update the total tenure
variable. Need to add positivity condition on the tenure
values since some of the values equal -3. */

do over match1;
if match1=3 and r4ten=>0 and r3ten1=>0 then do;
    totten=r3ten1+r4ten; end;
if match1=3 and -4<r3ten1<0 then do; totten=-3; end;
if match1=2 and r4ten=>0 and r2ten1=>0 then do;
    totten=r2ten1+r4ten; end;
if match1=2 and -4<r2ten1<0 then do; totten=-3; end;
if match1=1 and r4ten=>0 and r1ten1=>0 then do;
    totten=r1ten1+r4ten; end;
if match1=1 and -4<r1ten1<0 then do; totten=-3; end;
end;

/*repeat the above code for match2-match9*/

endsas;

```

TOTAL HOURS WORKED IN 19XX/20XX

Variables Created: CV_HOURS_WK_YR_ALL.80 – CV_HOURS_WK_YR_ALL.01
 CV_HOURS_WK_YR_ET.80 – CV_HOURS_WK_YR_ET.01
 CV_HOURS_WK_YR_SE.98 – CV_HOURS_WK_YR_SE.01

Programs Used

This program uses emp_begin.sas as input (see the first page of this appendix for details).

This program calculates the number of hours worked by the respondent at all jobs in each calendar year. Three sets of variables are created: one for all jobs, one for employee-type jobs, and one for self-employed jobs for respondents age 18 and older in round 4. See the introduction to this appendix for important information about the universes for each of these variables. A variable is created for each respondent even if the respondent has worked no jobs in a given year with the default value set to zero (0). Note that when both "starting hours" and "current hours" are reported, the latter are used to construct these measures.

```
*** Organize hours for each job;
array starw (k) starw1-starw9;
array stopw (k) stopw1-stopw9;

/* shrs1 - starting hours ehrs1 - ending hours */
array shrs (k) shrs1-shrs9;
array ehrs (k) ehrs1-ehrs9;
array hours (k) hours1-hours9;
array hrck (k) hrck1-hrck9;

/* To the right of the array statements are the true ranges of the pulled variables. Values of variables outside the ranges will be represented by dots. (this was done for simplicity for the programmer) */
array e23901 (k) e239011-e239019; /* 1-9 */ array e24501 (k) e245011-e245019; /* 1-6,9 */
array e34402 (k) e344021-e344029; /* 1-5 */ array e34403 (k) e344031-e344039; /* 1-3 */
array e34428 (k) e344281-e344289; /* 1-5 */ array e37904 (k) e379041-e379049; /* 1-5,7 */
array e38000f (k) e38000f1-e38000f9; /* 1-5,7 */ array e38103 (k) e381031-e381039; /* 1-3,5 */
array e38105 (k) e381051-e381059; /* 1-3 */ array e59901 (k) e599011-e599019; /* 1-7 */
array e88000 (k) e880001-e880009; /* 1-7 */ array e88501 (k) e885011-e885019; /* 1-5 */
array e98402 (k) e984021-e984209; /* 1-6 */ array e98403 (k) e984031-e984039; /* 1-3 */
array e98429 (k) e984291-e984299; /* 1-3 */ array e100231 (k) e1002311-e1002319; /* 1,2 */
array e100232 (k) e1002321-e1002329; /* 1,2 */ array e22610 (k) e226101-e226109; /* 1,2 */
array e22611 (k) e226111-e226119; /* 1,2 */ array starfl (k) starfl_1-starfl_9;
array stopfl (k) stopfl_1-stopfl_9;

/* Define the number of hours per week worked at the start date and the end date of a job. Question E59901 decides whether the job was listed on the Round 1 roster, for jobs under 13 weeks. Question E88000 also decides whether the job was listed on the Round 1 roster, but only for jobs that last longer than 13 weeks. The code below defines the starting hours and ending hours per week at those jobs, including overtime. */
do k=1 to 9;
/* Define starting hours for jobs shorter than 13 weeks */
if e23901>-4 then do; shrs=e23901; end;
if e23901>-4 and e24501>-4 then do; shrs=e23901+e24501; end;
if e34402>-4 then do; shrs=e34402; end;
if e34403>-4 then do; shrs=e34403; end;
if e34403>-4 and e34428>-4 then do; shrs=e34403+e34428; end;
/* Define ending hours for jobs less than 13 weeks */ if e37904>-4 then do; ehrs=e37904; end;
/* Define ending hours for jobs longer than 13 weeks */
if e38000f>-4 then do; ehrs=e38000f; end;
if e38103>-4 then do; ehrs=e38103; end;
if e38103>-4 and e38105>-4 then do; ehrs=e38103+e38105; end;
```

Appendix 2: Employment Variable Creation

```
if e59901>-4 then do; shrs=e59901; end; /* These are jobs from dli that are less than 13 weeks long */
if e88000>-4 then do; shrs=e88000; end; /* These are jobs from dli that are longer than 13 weeks */
if e88501>-4 and e88000>-4 then do; shrs=e88501+e88000; end; /* E88501 is an overtime question. */
if e98402>-4 then do; shrs=e98402; end;
if e98429>-4 then do; shrs=e98429; end;
if e98429>-4 and e98403>-4 then do; shrs=e98403+e98429; end;
if e22610>-4 then do; shrs=e22610; end;
if e22611>-4 then do; shrs=e22611; end;
if e22610>-4 and e22611>-4 then do; shrs=e22610+e22611; end;
if e100231>-4 then do; shrs=e100231; end;
if e100231>-4 then do; shrs=e100232; end;
if e100231>-4 and e100232>-4 then do; shrs=e100231+e100232; end;
end;

/* The code below decides which hours per week total (starting or ending) will be used in determining hours worked per
year. We prefer using ending hours as the measure, hours per week starts with a default number (-16), is
updated if starting hours are available, and is updated once more if ending hours are available.*/
* Set default hours to zero (-16);
do over hours; hours=0; hrck=0; end;

* Take starting hours if reported (eliminates -4's);
do over shrs;
if shrs>0 then do; hours=shrs; end;
/* hrck marks when invalid answers are given to the hrs/wk questions */
if -4<shrs<0 then do; hrck=shrs; end;
end;

* Write over if end hours reported;
do over ehrs;
if ehrs>0 then do; hours=ehrs; end;
if -4<ehrs<0 then do; hrck=ehrs; end;
end;

* Check for respondents who report valid starting hours and invalid ending hours...(11-29-01) There are 10 cases, all of
which have ehrs=-2. Since the respondent reports -2's, the c.v. will be -2 as well.:
do over ehrs;
if shrs>0 and -4<ehrs<0 then do; hrprob=1; end;
end;

/* This checks the people that report too many hours per week. */
do over hours; if hours>=60 then noway=1; end;

/* This fixes a person who has the job start & stop data at loop 4 and the hours data in loop three. Other than the hours
data, there is no other employment data in loop 3.*/
if id=1024861 then do; hours3=-.; hours4=14; end;

array job1wks (i) wk1_1-wk1_1148;      array job2wks (i) wk2_1-wk2_1148;      array job3wks (i) wk3_1-wk3_1148;
array job4wks (i) wk4_1-wk4_1148;      array job5wks (i) wk5_1-wk5_1148;      array job6wks (i) wk6_1-wk6_1148;
array job7wks (i) wk7_1-wk7_1148;      array job8wks (i) wk8_1-wk8_1148;      array job9wks (i) wk9_1-wk9_1148;

/** Calculate cumulative weeks on individual jobs for each year **/
/* 1980 */
array wks80 (i) wks801-wks809;      array swks80 (i) swks801-swks809;      array ewks80 (i) ewks801-ewks809;
array ah80 (i) ah801-ah809;          array sah80 (i) sah801-sah809;          array eah80 (i) eah801-eah809;

do i=1 to 9; wks80=0; swks80=0; ewks80=0; ah80=0; sah80=0; eah80=0; end;
```

Appendix 2: Employment Variable Creation

```

/* Update hours counter when respondent reports employment in weeks 1 to 52.*/
do i=1 to 52;
  if job1wks=1 then do; wks801=wks801+1; end;           if job2wks=1 then do; wks802=wks802+1; end;
  if job3wks=1 then do; wks803=wks803+1; end;           if job4wks=1 then do; wks804=wks804+1; end;
  if job5wks=1 then do; wks805=wks805+1; end;           if job6wks=1 then do; wks806=wks806+1; end;
  if job7wks=1 then do; wks807=wks807+1; end;           if job8wks=1 then do; wks808=wks808+1; end;
  if job9wks=1 then do; wks809=wks809+1; end;

  if job1wks=1 and self1=1 then do; swks801=swks801+1; end; /* Self-employed jobs */
  if job2wks=1 and self2=1 then do; swks802=swks802+1; end;
  /*and so on through*/
  if job9wks=1 and self9=1 then do; swks809=swks809+1; end;

  if job1wks=1 and self1=0 then do; ewks801=ewks801+1; end; /* Employee-type jobs */
  if job2wks=1 and self2=0 then do; ewks802=ewks802+1; end;
  /*and so on through*/
  if job9wks=1 and self9=0 then do; ewks809=ewks809+1; end;
end;

do i=1 to 52;
  if hours1>0 and job1wks ne -3 then ah801=hours1*wks801;
  if hours2>0 and job2wks ne -3 then ah802=hours2*wks802;
  /*and so on through*/
  if hours9>0 and job9wks ne -3 then ah809=hours9*wks809;

  if hours1>0 and job1wks ne -3 and self1=1 then sah801=hours1*swks801; /* Self-employed jobs */
  if hours2>0 and job2wks ne -3 and self2=1 then sah802=hours2*swks802;
  /*and so on through*/
  if hours9>0 and job9wks ne -3 and self9=1 then sah809=hours9*swks809;

  if hours1>0 and job1wks ne -3 and self1=0 then eah801=hours1*ewks801; /* Employee-type jobs */
  if hours2>0 and job2wks ne -3 and self2=0 then eah802=hours2*ewks802;
  /*and so on through*/
  if hours9>0 and job9wks ne -3 and self9=0 then eah809=hours9*ewks809;
end;

do i=1 to 52;
  if job1wks=-3 or 0<hours1<-4 then ah801=-3;          /* Define negative values for each job separately.*/
  if (job1wks=-3 or 0<hours1<-4) and self1=1 then sah801=-3;
  if (job1wks=-3 or 0<hours1<-4) and self1=0 then eah801=-3;
  if job2wks=-3 or 0<hours2<-4 then ah802=-3;
  if (job2wks=-3 or 0<hours2<-4) and self2=1 then sah802=-3;
  if (job2wks=-3 or 0<hours2<-4) and self2=0 then eah802=-3;
  /*and so on through*/
  if job9wks=-3 or 0<hours9<-4 then ah809=-3;
  if (job9wks=-3 or 0<hours9<-4) and self9=1 then sah809=-3;
  if (job9wks=-3 or 0<hours9<-4) and self9=0 then eah809=-3;
end;

/* Sum up the valid hours of each job */
if ah801=>0 and ah802=>0 and ah803=>0 and ah804=>0 and ah805=>0 and ah806=>0 and ah807=>0 and ah808=>0
  and ah809=>0 then do;
  r4hrs80=ah801+ah802+ah803+ah804+ah805+ah806+ah807+ah808+ah809;
end;
if sah801=>0 and sah802=>0 and sah803=>0 and sah804=>0 and sah805=>0 and sah806=>0 and sah807=>0 and
  sah808=>0 and sah809=>0 then do;
  r4shrs80=sah801+sah802+sah803+sah804+sah805+sah806+sah807+sah808+sah809;

```

Appendix 2: Employment Variable Creation

```

end;
if eah801=>0 and eah802=>0 and eah803=>0 and eah804=>0 and eah805=>0 and eah806=>0 and eah807=>0 and
    eah808=>0 and eah809=>0 then do;
    r4ehrs80=eah801+eah802+eah803+eah804+eah805+eah806+eah807+eah808+eah809;
end;

/* Place in invalid skips */
if ah801=-3 or ah802=-3 or ah803=-3 or ah804=-3 or ah805=-3 or ah806=-3 or ah807=-3 or ah808=-3 or ah809=-3 then
    do; r4hrs80=-3; end;
if sah801=-3 or sah802=-3 or sah803=-3 or sah804=-3 or sah805=-3 or sah806=-3 or sah807=-3 or sah808=-3 or
    sah809=-3 then do; r4shrs80=-3; end;
if eah801=-3 or eah802=-3 or eah803=-3 or eah804=-3 or eah805=-3 or eah806=-3 or eah807=-3 or eah808=-3 or
    eah809=-3 then do; r4ehrs80=-3; end;

/* At this point the program repeats the above 1980 code for each year through 2001. This code is omitted here due to
space considerations. Users who need to see the entire program should contact NLS User Services. The week
numbers and variables created for each year are as follows:

```

Year/ Weeks	Variables Created			Year/ Weeks	Variables Created		
1981 53-104	wks811-wks819;	ah811-ah819;		1992 627-678	wks921-wks929;	ah921-ah929;	
	swks811-swks819;	sah811-sah819;			swks921-swks929;	sah921-sah929;	
	ewks811-ewks819;	eah811-eah819;			ewks921-ewks929;	eah921-eah929;	
1982 105-156	wks821-wks829;	ah821-ah829;		1993 679-730	wks931-wks939;	ah931-ah939;	
	swks821-swks829;	sah821-sah829;			swks931-swks939;	sah931-sah939;	
	ewks821-ewks829;	eah821-eah829;			ewks931-ewks939;	eah931-eah939;	
1983 157-209	wks831-wks839;	ah831-ah839;		1994 731-783	wks941-wks949;	ah941-ah949;	
	swks831-swks839;	sah831-sah839;			swks941-swks949;	sah941-sah949;	
	ewks831-ewks839;	eah831-eah839;			ewks941-ewks949;	eah941-eah949;	
1984 210-261	wks841-wks849;	ah841-ah849;		1995 784-835	wks951-wks959;	ah951-ah959;	
	swks841-swks849;	sah841-sah849;			swks951-swks959;	sah951-sah959;	
	ewks841-ewks849;	eah841-eah849;			ewks951-ewks959;	eah951-eah959;	
1985 262-313	wks851-wks859;	ah851-ah859;		1996 836-887	wks961-wks969;	ah961-ah969;	
	swks851-swks859;	sah851-sah859;			swks961-swks969;	sah961-sah969;	
	ewks851-ewks859;	eah851-eah859;			ewks961-ewks969;	eah961-eah969;	
1986 314-365	wks861-wks869;	ah861-ah869;		1997 888-939	wks971-wks979;	ah971-ah979;	
	swks861-swks869;	sah861-sah869;			swks971-swks979;	sah971-sah979;	
	ewks861-ewks869;	eah861-eah869;			ewks971-ewks979;	eah971-eah979;	
1987 366-417	wks871-wks879;	ah871-ah879;		1998 940-991	wks981-wks989;	ah981-ah989;	
	swks871-swks879;	sah871-sah879;			swks981-swks989;	sah981-sah989;	
	ewks871-ewks879;	eah871-eah879;			ewks981-ewks989;	eah981-eah989;	
1988 418-470	wks881-wks889;	ah881-ah889;		1999 992-1043	wks991-wks999;	ah991-ah999;	
	swks881-swks889;	sah881-sah889;			swks991-swks999;	sah991-sah999;	
	ewks881-ewks889;	eah881-eah889;			ewks991-ewks999;	eah991-eah999;	
1989 471-522	wks891-wks899;	ah891-ah899;		2000 1044-1096	wks001-wks009;	ah001-ah009;	
	swks891-swks899;	sah891-sah899;			swks001-swks009;	sah001-sah009;	
	ewks891-ewks899;	eah891-eah899;			ewks001-ewks009;	eah001-eah009;	
1990 523-574	wks901-wks909;	ah901-ah909;		2001 1097-1148	wks011-wks019;	ah011-ah019;	
	swks901-swks909;	sah901-sah909;			swks011-swks019;	sah011-sah019;	
	ewks901-ewks909;	eah901-eah909;			ewks011-ewks019;	eah011-eah019;	
1991 575-626	wks911-wks919;	ah911-ah919;					
	swks911-swks919;	sah911-sah919;					
	ewks911-ewks919;	eah911-eah919;					

if e200 in (-3,-5) then do;

Appendix 2: Employment Variable Creation

```
r4hrs80=-5; r4hrs81=-5; r4hrs82=-5; r4hrs83=-5; r4hrs84=-5; r4hrs85=-5; r4hrs86=-5; r4hrs87=-5; r4hrs88=-5;
r4hrs89=-5; r4hrs90=-5; r4hrs91=-5; r4hrs92=-5; r4hrs93=-5; r4hrs94=-5; r4hrs95=-5; r4hrs96=-5; r4hrs97=-5;
r4hrs98=-5; r4hrs99=-5; r4hrs00=-5; r4hrs01=-5;
r4shrs80=-5; r4shrs81=-5; r4shrs82=-5; r4shrs83=-5; r4shrs84=-5; r4shrs85=-5; r4shrs86=-5; r4shrs87=-5; r4shrs88=-5;
r4shrs89=-5; r4shrs90=-5; r4shrs91=-5; r4shrs92=-5; r4shrs93=-5; r4shrs94=-5; r4shrs95=-5; r4shrs96=-5;
r4shrs97=-5; r4shrs98=-5; r4shrs99=-5; r4shrs00=-5; r4shrs01=-5;
r4ehrs80=-5; r4ehrs81=-5; r4ehrs82=-5; r4ehrs83=-5; r4ehrs84=-5; r4ehrs85=-5; r4ehrs86=-5; r4ehrs87=-5; r4ehrs88=-5;
r4ehrs89=-5; r4ehrs90=-5; r4ehrs91=-5; r4ehrs92=-5; r4ehrs93=-5; r4ehrs94=-5; r4ehrs95=-5; r4ehrs96=-5;
r4ehrs97=-5; r4ehrs98=-5; r4ehrs99=-5; r4ehrs00=-5; r4ehrs01=-5;
end;

/* Fill in -3's for cases where start/stop date is unknown/refused.*/
do k=1 to 9;

/* start date invalid */
if starfl=1 and tothrs80>0 and stopw>1 then do; tothrs80=-3; end;
if starfl=1 and tothrs81>0 and stopw>52 then do; tothrs81=-3; end;
if starfl=1 and tothrs82>0 and stopw>104 then do; tothrs82=-3; end;
if starfl=1 and tothrs83>0 and stopw>156 then do; tothrs83=-3; end;
if starfl=1 and tothrs84>0 and stopw>209 then do; tothrs84=-3; end;
if starfl=1 and tothrs85>0 and stopw>261 then do; tothrs85=-3; end;
if starfl=1 and tothrs86>0 and stopw>313 then do; tothrs86=-3; end;
if starfl=1 and tothrs87>0 and stopw>365 then do; tothrs87=-3; end;
if starfl=1 and tothrs88>0 and stopw>417 then do; tothrs88=-3; end;
if starfl=1 and tothrs89>0 and stopw>470 then do; tothrs89=-3; end;
if starfl=1 and tothrs90>0 and stopw>522 then do; tothrs90=-3; end;
if starfl=1 and tothrs91>0 and stopw>574 then do; tothrs91=-3; end;
if starfl=1 and tothrs92>0 and stopw>626 then do; tothrs92=-3; end;
if starfl=1 and tothrs93>0 and stopw>678 then do; tothrs93=-3; end;
if starfl=1 and tothrs94>0 and stopw>730 then do; tothrs94=-3; end;
if starfl=1 and tothrs95>0 and stopw>783 then do; tothrs95=-3; end;
if starfl=1 and tothrs96>0 and stopw>835 then do; tothrs96=-3; end;
if starfl=1 and tothrs97>0 and stopw>887 then do; tothrs97=-3; end;
if starfl=1 and tothrs98>0 and stopw>939 then do; tothrs98=-3; end;
if starfl=1 and tothrs99>0 and stopw>990 then do; tothrs99=-3; end;
if starfl=1 and tothrs00>0 and stopw>1043 then do; tothrs00=-3; end;
if starfl=1 and tothrs01>0 and stopw>1096 then do; tothrs01=-3; end;

/* stop date invalid */
if stopfl=1 and tothrs80>0 and starw<53 then do; tothrs80=-3; end;
if stopfl=1 and tothrs81>0 and starw<105 then do; tothrs81=-3; end;
if stopfl=1 and tothrs82>0 and starw<157 then do; tothrs82=-3; end;
if stopfl=1 and tothrs83>0 and starw<210 then do; tothrs83=-3; end;
if stopfl=1 and tothrs84>0 and starw<262 then do; tothrs84=-3; end;
if stopfl=1 and tothrs85>0 and starw<314 then do; tothrs85=-3; end;
if stopfl=1 and tothrs86>0 and starw<366 then do; tothrs86=-3; end;
if stopfl=1 and tothrs87>0 and starw<418 then do; tothrs87=-3; end;
if stopfl=1 and tothrs88>0 and starw<471 then do; tothrs88=-3; end;
if stopfl=1 and tothrs89>0 and starw<523 then do; tothrs89=-3; end;
if stopfl=1 and tothrs90>0 and starw<575 then do; tothrs90=-3; end;
if stopfl=1 and tothrs91>0 and starw<627 then do; tothrs91=-3; end;
if stopfl=1 and tothrs92>0 and starw<679 then do; tothrs92=-3; end;
if stopfl=1 and tothrs93>0 and starw<731 then do; tothrs93=-3; end;
if stopfl=1 and tothrs94>0 and starw<784 then do; tothrs94=-3; end;
if stopfl=1 and tothrs95>0 and starw<836 then do; tothrs95=-3; end;
if stopfl=1 and tothrs96>0 and starw<888 then do; tothrs96=-3; end;
if stopfl=1 and tothrs97>0 and starw<940 then do; tothrs97=-3; end;
```

```

if stopfl=1 and tothrs98>0 and starw<991 then do; tothrs98=-3; end;
if stopfl=1 and tothrs99>0 and starw<1044 then do; tothrs99=-3; end;
if stopfl=1 and tothrs00>0 and starw<1096 then do; tothrs00=-3; end;
if stopfl=1 and tothrs01>0 and starw<1149 then do; tothrs01=-3; end;
end;

*****this section merges the round 4 data with the data from previous rounds*****
array thrs (i) thrs80-thrs99 thrs00 thrs01; array tshrs (i) tshrs80-tshrs99 tshrs00 tshrs01;
array tehrs (i) tehrs80-tehrs99 tehrs00 tehrs01; array r1hrs (i) r1hrs80-r1hrs99 r1hrs00 r1hrs01;
array r2hrs (i) r2hrs80-r2hrs99 r2hrs00 r2hrs01; array r3hrs (i) r3hrs80-r3hrs99 r3hrs00 r3hrs01;
array r4hrs (i) r4hrs80-r4hrs99 r4hrs00 r4hrs01; array r4shrs (i) r4shrs80-r4shrs99 r4shrs00 r4shrs01;
array r4ehrs (i) r4ehrs80-r4ehrs99 r4ehrs00 r4ehrs01;

/* Begin by splitting up periods where Round 4 and dli Round exclusively collect hours worked information. Any hours
   worked information collected before the dli Round interview year should be independent of information
   collected in Round 4. For the new Round 4 c.v.'s, we assume that all dli jobs are employee-type jobs. Thus, dli
   information will be added to the "all jobs" and "employee jobs" c.v.'s but not "self-employed jobs" c.v.'s. */
do over r4hrs;
    thrs=0; tshrs=0; tehrs=0;
    if r4hrs>0 then do; thrs=r4hrs; tehrs=r4ehrs; end;
    if r4shrs>0 then do; tshrs=r4shrs; end;
    if r3hrs>0 and dliwk=r3int then do; thrs=r3hrs; tehrs=r3hrs; end;
    if r2hrs>0 and dliwk=r2int then do; thrs=r2hrs; tehrs=r2hrs; end;
    if r1hrs>0 and dliwk=r1int then do; thrs=r1hrs; tehrs=r1hrs; end;
    if r3hrs>0 and r4hrs>0 and dliwk=r3int then do; thrs=r3hrs+r4hrs; end;
    if r3hrs>0 and r4ehrs>0 and dliwk=r3int then do; tehrs=r3hrs+r4ehrs; end;
    if r2hrs>0 and r4hrs>0 and dliwk=r2int then do; thrs=r2hrs+r4hrs; end;
    if r2hrs>0 and r4ehrs>0 and dliwk=r2int then do; tehrs=r2hrs+r4ehrs; end;
    if r1hrs>0 and r4hrs>0 and dliwk=r1int then do; thrs=r1hrs+r4hrs; end;
    if r1hrs>0 and r4ehrs>0 and dliwk=r1int then do; tehrs=r1hrs+r4ehrs; end;
end;

do over r4hrs;
    if -4<r4hrs<0 then do; thrs=r4hrs; end;
    if -4<r4shrs<0 then do; tshrs=r4shrs; end;
    if -4<r4ehrs<0 then do; tehrs=r4ehrs; end;
    if -4<r3hrs<0 and dliwk=r3int then do; thrs=r3hrs; tehrs=r3hrs; end;
    if -4<r2hrs<0 and dliwk=r2int then do; thrs=r2hrs; tehrs=r2hrs; end;
    if -4<r1hrs<0 and dliwk=r1int then do; thrs=r1hrs; tehrs=r1hrs; end;
end;

do over thrs;
    if e200=-5 then do; thrs=-5; tehrs=-5; tshrs=-5; end;
end;

endsas;

```

TOTAL HOURS WORKED AGE 14-19, HOURS WORKED AT ALL JOBS SINCE AGE 20, AND HOURS WORKED AT EMPLOYEE-TYPE JOBS SINCE AGE 20

Variables Created: CV_HOURS_WK_TEEN
CV_HOURS_WK_ADULT_ALL
CV_HOURS_WK_ADULT_ET

Programs Used

This program uses `emp_begin.sas` as input (see the first page of this appendix for details).

This program calculates the number of hours worked by the respondent at all employee-type jobs between the ages of 14 and 19. It then creates two variables for the number of hours worked by the respondent since age 20: one for all jobs and one for employee-type jobs. See the introduction to this appendix for important information about the universes for each of these variables. A variable is created for each respondent even if the respondent has worked no jobs in a given year with the default value set to zero (0). Note that when both "starting hours" and "current hours" are reported, the latter are used to construct these measures.

<pre>*** Organize hours for each job; array starw (i) starw1-starw9; array stopw (i) stopw1-stopw9; array starfl (i) starfl_1-starfl_9; array stopfl (i) stopfl_1-stopfl_9; /* shrs1 - starting hours ehrs1 - ending hours */ array shrs (i) shrs1-shrs9; array ehrs (i) ehrs1-ehrs9; array hours (i) hours1-hours9; array hrck (i) hrck1-hrck9; /* To the right of the array statements are the true ranges of the pulled variables. Values of variables outside the ranges will be represented by dots. (this was done for simplicity for the programmer) */ array e22610 (i) e226101-e226109; array e22611 (i) e226111-e226119; array e100231 (i) e1002311-e1002319; array e100232 (i) e1002321-e1002329; array e23901 (i) e239011-e239019; /* 1-9 */ array e24501 (i) e245011-e245019; /* 1-6,9 */ array e34402 (i) e344021-e344029; /* 1-5 */ array e34403 (i) e344031-e344039; /* 1-3 */ array e34428 (i) e344281-e344289; /* 1-5 */ array e37904 (i) e379041-e379049; /* 1-5,7 */ array e38000f (i) e38000f1-e38000f9; /* 1-5,7 */ array e38103 (i) e381031-e381039; /* 1-3,5 */ array e38105 (i) e381051-e381059; /* 1-3 */ array e59901 (i) e599011-e599019; /* 1-7 */ array e88000 (i) e880001-e880009; /* 1-7 */ array e88501 (i) e885011-e885019; /* 1-5 */ array e98402 (i) e984021-e984209; /* 1-6 */ array e98403 (i) e984031-e984039; /* 1-3 */ array e98429 (i) e984291-e984299; /* 1-3 */ array e100256 (i) e1002561-e1002569; /* 1,2 */</pre>	<pre>/* Define the number of hours per week worked at the start date and the end date of a job. Question E59901 decides whether the job was listed on the Round 1 roster, for jobs under 13 weeks. Question E88000 also decides whether the job was listed on the Round 1 roster, but only for jobs that last longer than 13 weeks. The code below defines the starting hours and ending hours per week at those jobs, including overtime. */ do i=1 to 9; /* Define starting hours for jobs shorter than 13 weeks */ if e23901>-4 then do; shrs=e23901; end; if e23901>-4 and e24501>-4 then do; shrs=e23901+e24501; end; if e34402>-4 then do; shrs=e34402; end; if e34403>-4 then do; shrs=e34403; end; if e34403>-4 and e34428>-4 then do; shrs=e34403+e34428; end; /* Define ending hours for jobs less than 13 weeks long */ if e37904>-4 then do; ehrs=e37904; end; /* Define ending hours for jobs longer than 13 weeks */ if e38000f>-4 then do; ehrs=e38000f; end; if e38103>-4 then do; ehrs=e38103; end; if e38103>-4 and e38105>-4 then do; ehrs=e38103+e38105; end; /* These are jobs from dli less than 13 weeks long*/ if e59901>-4 then do; shrs=e59901; end; /* These are jobs from dli that are longer than 13 weeks */ if e88000>-4 then do; shrs=e88000; end; /* E88501 is an overtime question. */ if e88501>-4 and e88000>-4 then do; shrs=e88501+e88000; end; if e98402>-4 then do; shrs=e98402; end; if e98429>-4 then do; shrs=e98429; end; if e98429>-4 and e98403>-4 then do; shrs=e98403+e98429; end; if e22610>-4 then do; shrs=e22610; end; if e22611>-4 then do; shrs=e22611; end;</pre>
---	--

```

if e22610>-4 and e22611>-4 then do;
    shrs=e22610+e22611; end;
if e100231>-4 then do; shrs=e100231; end;
if e100231>-4 then do; shrs=e100232; end;
if e100231>-4 and e100232>-4 then do;
    shrs=e100231+e100232; end;
end;

/* The code below decides which hours per week total
(starting or ending) will be used in determining hours
worked per year. We prefer using ending hours as the
measure, hours per week starts with a default number (-16),
is updating if starting hours are available, and is
updated once more if ending hours are available. */

* Set default hours to zero (-16);
do over hours; hours=0; hrck=0; end;

* Take starting hours if reported (eliminates -4's);
do over shrs;
    if shrs>=0 then do; hours=shrs; end;
/* hrck marks invalid answers for the hrs/wk questions */
    if -4<shrs<0 then do; hrck=shrs; end;
end;

* Write over if end hours reported;
do over ehrs;
    if ehrs>=0 then do; hours=ehrs; end;
    if -4<ehrs<0 then do; hrck=ehrs; end;
end;

array job1wks (i) wk1_1-wk1_1148;
array job2wks (i) wk2_1-wk2_1148;
array job3wks (i) wk3_1-wk3_1148;
array job4wks (i) wk4_1-wk4_1148;
array job5wks (i) wk5_1-wk5_1148;
array job6wks (i) wk6_1-wk6_1148;
array job7wks (i) wk7_1-wk7_1148;
array job8wks (i) wk8_1-wk8_1148;
array job9wks (i) wk9_1-wk9_1148;

/** Calculate cumulative weeks on employee-type jobs for
each year **/

/* Initialize created variables */
r4hrs14=0; r4hrs20=0; r4ehrs20=0;

/* Read "wks145" as the number of weeks worked
between the ages of 14 and 19, for job 5 only. Read
"thrs145" as the number of hours worked between the
ages of 14 and 19 for job 5 only. */
wks141=0; wks142=0; wks143=0; wks144=0; wks145=0;
wks146=0; wks147=0; wks148=0; wks149=0; thrs141=0;
thrs142=0; thrs143=0; thrs144=0; thrs145=0; thrs146=0;
thrs147=0; thrs148=0; thrs149=0;

/* Respondents who have not turned 20 by Round 4. */

```

```

if age14wk>0 and (age20wk-1)>1148 then do;
    do i=age14wk to 1148;
        if job1wks=1 and self1=0 then do; wks141=wks141+1;
            end;
        if job2wks=1 and self2=0 then do; wks142=wks142+1;
            end;
        /*and so on through*/
        if job9wks=1 and self9=0 then do; wks149=wks149+1;
            end;
    end;

do i=age14wk to 1148;
    if job1wks ne -3 and hours1=>0 and self1=0 then
        thrs141=hours1*wks141;
    if job2wks ne -3 and hours2=>0 and self2=0 then
        thrs142=hours2*wks142;
    /*and so on through*/
    if job9wks ne -3 and hours9=>0 and self9=0 then
        thrs149=hours9*wks149;
end;

do i=age14wk to 1148;
    if (job1wks=-3 and self1=0) then thrs141=-3;
    if (job2wks=-3 and self2=0) then thrs142=-3;
    /*and so on through*/
    if (job9wks=-3 and self9=0) then thrs149=-3;
end;

if thrs141=>0 and thrs142=>0 and thrs143=>0 and
    thrs144=>0 and thrs145=>0 and thrs146=>0 and
    thrs147=>0 and thrs148=>0 and thrs149=>0 then
    do;
        r4hrs14=thrs141 + thrs142 + thrs143 + thrs144 +
            thrs145 + thrs146 + thrs147 + thrs148 + thrs149;
    end;
if (thrs141=-3 and self1=0) or (thrs142=-3 and self2=0) or
    (thrs143=-3 and self3=0) or (thrs144=-3 and
        self4=0) or (thrs145=-3 and self5=0) or (thrs146=-3
            and self6=0) or (thrs147=-3 and self7=0) or
                (thrs148=-3 and self8=0) or (thrs149=-3 and
                    self9=0) then do;
    r4hrs14=-3;
end;

/* hrck is used when invalid hrs/wk answers are given */
if wks141>0 and hrck1<0 and r4hrs14>0 and self1=0 then
    r4hrs14=-3;
if wks142>0 and hrck2<0 and r4hrs14>0 and self2=0 then
    r4hrs14=-3;
/*and so on through */
if wks149>0 and hrck9<0 and r4hrs14>0 and self9=0 then
    r4hrs14=-3;
end;

/* Respondents who have turned 20 by Round 3. */
if age14wk>0 and (age20wk-1)<=1148 then do;
    do i=age14wk to (age20wk-1);

```

Appendix 2: Employment Variable Creation

```

if job1wks=1 and self1=0 then do; wks141=wks141+1;
    end;
if job2wks=1 and self2=0 then do; wks142=wks142+1;
    end;
/*and so on through*/
if job9wks=1 and self9=0 then do; wks149=wks149+1;
    end;
end;

do i=age14wk to (age20wk-1);
    if job1wks ne -3 and hours1=>0 and self1=0 then
        thrs141=hours1*wks141;
    if job2wks ne -3 and hours2=>0 and self2=0 then
        thrs142=hours2*wks142;
    /*and so on through*/
    if job9wks ne -3 and hours9=>0 and self9=0 then
        thrs149=hours9*wks149;
end;

do i=age14wk to (age20wk-1);
    if (job1wks=-3 and self1=0) then thrs141=-3;
    if (job2wks=-3 and self2=0) then thrs142=-3;
    /*and so on through*/
    if (job9wks=-3 and self9=0) then thrs149=-3;
end;

if thrs141=>0 and thrs142=>0 and thrs143=>0 and
    thrs144=>0 and thrs145=>0 and thrs146=>0 and
    thrs147=>0 and thrs148=>0 and thrs149=>0 then
    do;
        r4hrs14=thrs141 + thrs142 + thrs143 + thrs144 +
            thrs145 + thrs146 + thrs147 + thrs148 + thrs149;
    end;
if (thrs141=-3 and self1=0) or (thrs142=-3 and self2=0)
    or (thrs143=-3 and self3=0) or (thrs144=-3 and
        self4=0) or (thrs145=-3 and self5=0) or (thrs146=-3
        and self6=0) or (thrs147=-3 and self7=0) or
        (thrs148=-3 and self8=0) or (thrs149=-3 and
        self9=0) then do;
    r4hrs14=-3;
end;

/* hrck is used when invalid hrs./wk answers are given */
if wks141>0 and hrck1<0 and r4hrs14>0 and self1=0
    then r4hrs14=-3;
if wks142>0 and hrck2<0 and r4hrs14>0 and self2=0
    then r4hrs14=-3;
/*and so on through*/
if wks149>0 and hrck9<0 and r4hrs14>0 and self9=0
    then r4hrs14=-3;
end;

/* Read "wks205" as the number of weeks worked after
age 20 for job 5 only. Read "thrs205" as the number of
hours worked after age 20 for job 5 only. */
wks201=0; wks202=0; wks203=0; wks204=0; wks205=0;
wks206=0; wks207=0; wks208=0; wks209=0;

thrs201=0; thrs202=0; thrs203=0; thrs204=0; thrs205=0;
thrs206=0; thrs207=0; thrs208=0; thrs209=0;

/* Respondents who have turned 20 by Round 3. */
if age14wk>0 and (age20wk-1)<=1148 then do;
    do i=age20wk to 1148;
        if job1wks=1 then do; wks201=wks201+1; end;
        if job2wks=1 then do; wks202=wks202+1; end;
        /*and so on through*/
        if job9wks=1 then do; wks209=wks209+1; end;
    end;

    do i=age20wk to 1148;
        if job1wks ne -3 and hours1=>0 then
            thrs201=hours1*wks201;
        if job2wks ne -3 and hours2=>0 then
            thrs202=hours2*wks202;
        /*and so on through*/
        if job9wks ne -3 and hours9=>0 then
            thrs209=hours9*wks209;
    end;

    do i=age20wk to 1148;
        if (job1wks=-3) then thrs201=-3;
        if (job2wks=-3) then thrs202=-3;
        /*and so on through*/
        if (job9wks=-3) then thrs209=-3;
    end;

if thrs201=>0 and thrs202=>0 and thrs203=>0 and
    thrs204=>0 and thrs205=>0 and thrs206=>0 and
    thrs207=>0 and thrs208=>0 and thrs209=>0 then
    do;
        r4hrs20=thrs201 + thrs202 + thrs203 + thrs204 +
            thrs205 + thrs206 + thrs207 + thrs208 + thrs209;
    end;
if thrs201=-3 or thrs202=-3 or thrs203=-3 or thrs204=-3
    or thrs205=-3 or thrs206=-3 or thrs207=-3 or
    thrs208=-3 or thrs209=-3 then do;
    r4hrs20=-3;
end;

/* hrck is used when invalid hrs./wk answers are given */
if wks201>0 & hrck1<0 & r4hrs20>0 then r4hrs20=-3;
if wks202>0 & hrck2<0 & r4hrs20>0 then r4hrs20=-3;
/*and so on through*/
if wks209>0 & hrck9<0 & r4hrs20>0 then r4hrs20=-3;
end;

/* Read "wks205" as the number of weeks worked after
age 20 for job 5 only. Read "thrs205" as the number of
hours worked after age 20 for job 5 only. */
ewks201=0; ewks202=0; ewks203=0; ewks204=0;
ewks205=0; ewks206=0; ewks207=0; ewks208=0;
ewks209=0;

```

```

tehrs201=0; tehrs202=0; tehrs203=0; tehrs204=0;
tehrs205=0; tehrs206=0; tehrs207=0; tehrs208=0;
tehrs209=0;

/* Respondents who have turned 20 by Round 3. */
if age14wk>0 and (age20wk-1)<=1148 then do;
do i=age20wk to 1148;
  if job1wks=1 and self1=0 then do;
    ewks201=ewks201+1; end;
  if job2wks=1 and self2=0 then do;
    ewks202=ewks202+1; end;
    /*and so on through*/
  if job9wks=1 and self9=0 then do;
    ewks209=ewks209+1; end;
  end;

  do i=age20wk to 1148;
    if job1wks ne -3 and hours1=>0 and self1=0 then
      tehrs201=hours1*ewks201;
    if job2wks ne -3 and hours2=>0 and self2=0 then
      tehrs202=hours2*ewks202;
    /*and so on through*/
    if job9wks ne -3 and hours9=>0 and self9=0 then
      tehrs209=hours9*ewks209;
  end;

  do i=age20wk to 1148;
    if (job1wks=-3 and self1=0) then tehrs201=-3;
    if (job2wks=-3 and self2=0) then tehrs202=-3;
    /*and so on through*/
    if (job9wks=-3 and self9=0) then tehrs209=-3;
  end;

if tehrs201=>0 and tehrs202=>0 and tehrs203=>0 and
  tehrs204=>0 and tehrs205=>0 and tehrs206=>0
  and tehrs207=>0 and tehrs208=>0 and
  tehrs209=>0 then do;
  r4ehrs20=tehrs201+tehrs202+tehrs203+tehrs204+
    tehrs205+tehrs206+tehrs207+tehrs208+tehrs209;
end;

if tehrs201=-3 or tehrs202=-3 or tehrs203=-3 or
  tehrs204=-3 or tehrs205=-3 or tehrs206=-3 or
  tehrs207=-3 or tehrs208=-3 or tehrs209=-3 then
do;
  r4ehrs20=-3;
end;

/* hrck is used when invalid hrs./wk answers are given */
if ewks201>0 and hrck1<0 and r4ehrs20>0 and self1=0
  then r4ehrs20=-3;
if ewks202>0 and hrck2<0 and r4ehrs20>0 and self2=0
  then r4ehrs20=-3;
  /*and so on through*/
if ewks209>0 and hrck9<0 and r4ehrs20>0 and self9=0
  then r4ehrs20=-3;
end;

```

```

array self (i) self1-self9;

/* Remove invalid start/stop dates */
do i=1 to 9;
  if age14wk<784 then do; if starfl=1 and r4hrs14>0 and
    stopw>730 then do; r4hrs14=-3; end; end;
  if age14wk<836 then do; if starfl=1 and r4hrs14>0 and
    stopw>783 then do; r4hrs14=-3; end; end;
  if age14wk<888 then do; if starfl=1 and r4hrs14>0 and
    stopw>835 then do; r4hrs14=-3; end; end;
  if age14wk<940 then do; if starfl=1 and r4hrs14>0 and
    stopw>887 then do; r4hrs14=-3; end; end;
  if age14wk<991 then do; if starfl=1 and r4hrs14>0 and
    stopw>939 then do; r4hrs14=-3; end; end;
  if age14wk<1044 then do; if starfl=1 and r4hrs14>0 and
    stopw>990 then do; r4hrs14=-3; end; end;
  if age14wk<1097 then do; if starfl=1 and r4hrs14>0 and
    stopw>1043 then do; r4hrs14=-3; end; end;
  if age20wk<1097 then do; if starfl=1 and r4hrs20>0 and
    stopw>1043 then do; r4hrs20=-3; end; end;
  if age20wk<1097 then do; if starfl=1 and self=0 and
    r4ehrs20>0 and stopw>1043 then do; r4ehrs20=-3;
  end; end;
  if age14wk<1149 then do; if starfl=1 and r4hrs14>0 and
    stopw>1096 then do; r4hrs14=-3; end; end;
  if age20wk<1149 then do; if starfl=1 and r4hrs20>0 and
    stopw>1096 then do; r4hrs20=-3; end; end;
  if age20wk<1149 then do; if starfl=1 and self=0 and
    r4ehrs20>0 and stopw>1096 then do; r4ehrs20=-3;
  end; end;

  if age14wk<784 then do; if stopfl=1 and r4hrs14>0 and
    starw<784 then do; r4hrs14=-3; end; end;
  if age14wk<836 then do; if stopfl=1 and r4hrs14>0 and
    starw<836 then do; r4hrs14=-3; end; end;
  if age14wk<888 then do; if stopfl=1 and r4hrs14>0 and
    starw<888 then do; r4hrs14=-3; end; end;
  if age14wk<940 then do; if stopfl=1 and r4hrs14>0 and
    starw<940 then do; r4hrs14=-3; end; end;
  if age14wk<991 then do; if stopfl=1 and r4hrs14>0 and
    starw<991 then do; r4hrs14=-3; end; end;
  if age14wk<1044 then do; if stopfl=1 and r4hrs14>0 and
    starw<1044 then do; r4hrs14=-3; end; end;
  if age14wk<1097 then do; if stopfl=1 and r4hrs14>0 and
    starw<1097 then do; r4hrs14=-3; end; end;
  if age20wk<1097 then do; if stopfl=1 and r4hrs20>0 and
    starw<1097 then do; r4hrs20=-3; end; end;
  if age20wk<1097 then do; if stopfl=1 and self=0 and
    r4ehrs20>0 and starw<1097 then do; r4ehrs20=-3;
  end; end;
  if age14wk<1149 then do; if stopfl=1 and r4hrs14>0 and
    starw<1149 then do; r4hrs14=-3; end; end;
  if age20wk<1149 then do; if stopfl=1 and r4hrs20>0 and
    starw<1149 then do; r4hrs20=-3; end; end;

```

Appendix 2: Employment Variable Creation

```

if age20wk<1149 then do; if stopfl=1 and self=0 and
   r4ehrs20>0 and starw<1149 then do; r4ehrs20=-3;
   end; end;
end;

if e200=-5 then do;
   r4hrs14=-5; r4hrs20=-5; r4ehrs20=-5;
end;

r3hrs14=tohrs14;
r3hrs20=tohrs20;
drop tohrs14 tohrs20;

/* Initialize created variables to zero. */
thrs14=0; thrs20=0; tehrs20=0;

/* By the construction of the dli created variables, we can
simply add the two created variables from the current and
previous rounds if they are both positive. If one is positive
and one is not, then the positive value will be the total
created variable for both rounds. If neither is positive,
then the total created variable for both rounds will be 0.*/
if r4hrs14=>0 then do; thrs14=r4hrs14; end;
if r4hrs14=-3 then do; thrs14=-3; end;

/* dli is Round 3 */
if r3hrs14=>0 and r4hrs14=>0 then do;
   thrs14=r3hrs14+r4hrs14; end;
if r3hrs14=-3 then do; thrs14=-3; end;

/* dli is Round 2 */
if r2hrs14=>0 and r4hrs14=>0 and r3int=-5 then do;
   thrs14=r2hrs14+r4hrs14; end;

if r2hrs14=-3 and r3int=-5 then do; thrs14=-3; end;

/* dli is Round 1 */
if r1hrs14=>0 and r4hrs14=>0 and r3int=-5 and r2int=-5
   then do; thrs14=r1hrs14+r4hrs14; end;
if r1hrs14=-3 and r3int=-5 and r2int=-5 then do; thrs14=-3; end;

if r4hrs20>0 then thrs20=r4hrs20;
if r3hrs20>0 then thrs20=r3hrs20;
if r4hrs20>0 and r3hrs20>0 then do;
   thrs20=r4hrs20+r3hrs20; end;
if r4hrs20=-3 or r3hrs20=-3 then do; thrs20=-3; end;

if r4ehrs20>0 then tehrs20=r4ehrs20;
if r3hrs20>0 then tehrs20=r3hrs20;
if r4ehrs20>0 and r3hrs20>0 then do;
   tehrs20=r4ehrs20+r3hrs20; end;
if r4ehrs20=-3 or r3hrs20=-3 then do; tehrs20=-3; end;

/* For respondents younger than 20 at interview date,
   their thrs20 c.v. will be -4. */
if age20wk>intwk and intwk>0 then do; thrs20=-4;
   tehrs20=-4; end;

/* Account for non-interview cases */
if uid1=-5 then do;
   thrs14=-5; thrs20=-5; tehrs20=-5;
end;

endsas;

```

NUMBER OF JOBS HELD DURING 19XX/20XX

Variables Created: CV_TTL_JOB_ALL.80 – CV_TTL_JOB_ALL.01
 CV_TTL_JOB_ET.80 – CV_TTL_JOB_ET.01
 CV_TTL_JOB_SE.98 – CV_TTL_JOB_SE.01

Programs Used

This program uses `emp_begin.sas` as input (see the first page of this appendix for details).

This program calculates the number of employee-type jobs the respondent held during each calendar year. Three sets of variables are created: one for all jobs, one for employee-type jobs, and one for self-employed jobs. **See the introduction to this appendix for important information about the universes for each of these variables.** These variables are created only for respondents who have worked at least one week in the applicable calendar year.

```

array job1wks (i) wk1_1-wk1_1148;
array job2wks (i) wk2_1-wk2_1148;
array job3wks (i) wk3_1-wk3_1148;
array job4wks (i) wk4_1-wk4_1148;
array job5wks (i) wk5_1-wk5_1148;
array job6wks (i) wk6_1-wk6_1148;
array job7wks (i) wk7_1-wk7_1148;
array job8wks (i) wk8_1-wk8_1148;
array job9wks (i) wk9_1-wk9_1148;
array starw (i) starw1-starw9;
array stopw (i) stopw1-stopw9;
array starfl (i) starfl_1-starfl_9;
array stopfl (i) stopfl_1-stopfl_9;

/** Indicate if worked at least one week on a job in a
given year ** /

/* 1980 */
r4j801=0; r4s801=0; r4e801=0;
r4j802=0; r4s802=0; r4e802=0;
r4j803=0; r4s803=0; r4e803=0;
r4j804=0; r4s804=0; r4e804=0;
r4j805=0; r4s805=0; r4e805=0;
r4j806=0; r4s806=0; r4e806=0;
r4j807=0; r4s807=0; r4e807=0;
r4j808=0; r4s808=0; r4e808=0;
r4j809=0; r4s809=0; r4e809=0;

do i=1 to 52;
    if job1wks=-3 then do; r4j801=-3; end;
    if job1wks=-3 and self1=1 then do; r4s801=-3; end;
    if job1wks=-3 and self1=0 then do; r4e801=-3; end;
    if job2wks=-3 then do; r4j802=-3; end;
    if job2wks=-3 and self2=1 then do; r4s802=-3; end;
    if job2wks=-3 and self2=0 then do; r4e802=-3; end;
    /*and so on through*/
    if job9wks=-3 then do; r4j809=-3; end;
    if job9wks=-3 and self9=1 then do; r4s809=-3; end;
    if job9wks=-3 and self9=0 then do; r4e809=-3; end;
end;

do i=1 to 52;

```

```

if job1wks=1 then do; r4j801=1; end;
if job1wks=1 and self1=1 then do; r4s801=1; end;
if job1wks=1 and self1=0 then do; r4e801=1; end;
if job2wks=1 then do; r4j802=1; end;
if job2wks=1 and self2=1 then do; r4s802=1; end;
if job2wks=1 and self2=0 then do; r4e802=1; end;
/*and so on through*/
if job9wks=1 then do; r4j809=1; end;
if job9wks=1 and self9=1 then do; r4s809=1; end;
if job9wks=1 and self9=0 then do; r4e809=1; end;
end;

if r4j801 ne -3 and r4j802 ne -3 and r4j803 ne -3 and
   r4j804 ne -3 and r4j805 ne -3 and r4j806 ne -3 and
   r4j807 ne -3 and r4j808 ne -3 and r4j809 ne -3
   then do;
   r4job80=sum(r4j801, r4j802, r4j803, r4j804, r4j805,
   r4j806, r4j807, r4j808, r4j809);
end;
if r4j801=-3 or r4j802=-3 or r4j803=-3 or r4j804=-3 or
   r4j805=-3 or r4j806=-3 or r4j807=-3 or r4j808=-3
   or r4j809=-3 then do;
   r4job80=-3;
end;

if r4s801 ne -3 and r4s802 ne -3 and r4s803 ne -3 and
   r4s804 ne -3 and r4s805 ne -3 and r4s806 ne -3 and
   r4s807 ne -3 and r4s808 ne -3 and r4s809 ne -3
   then do;
   r4self80=sum(r4s801, r4s802, r4s803, r4s804, r4s805,
   r4s806, r4s807, r4s808, r4s809);
end;
if r4s801=-3 or r4s802=-3 or r4s803=-3 or r4s804=-3 or
   r4s805=-3 or r4s806=-3 or r4s807=-3 or r4s808=-3
   or r4s809=-3 then do;
   r4self80=-3;
end;

if r4e801 ne -3 and r4e802 ne -3 and r4e803 ne -3 and
   r4e804 ne -3 and r4e805 ne -3 and r4e806 ne -3
   and r4e807 ne -3 and r4e808 ne -3 and r4e809 ne -
   3 then do;

```

Appendix 2: Employment Variable Creation

```

r4emp80=sum(r4e801, r4e802, r4e803, r4e804,
             r4e805, r4e806, r4e807, r4e808, r4e809);
end;
if r4e801=-3 or r4e802=-3 or r4e803=-3 or r4e804=-3 or
   r4e805=-3 or r4e806=-3 or r4e807=-3 or r4e808=-3
   or r4e809=-3 then do;
  r4emp80=-3;
end;

/* At this point the program repeats the above code for
each year through 2001. The code is omitted here due to
space considerations. Users should contact NLS User
Services for more information. The week numbers for the
"do i" statements are as follows:
  1981  do i=53 to 104    1992  do i=627 to 679
  1982  do i=105 to 156   1993  do i=679 to 730
  1983  do i=157 to 209   1994  do i=731 to 783
  1984  do i=210 to 261   1995  do i=784 to 835
  1985  do i=262 to 313   1996  do i=836 to 887
  1986  do i=314 to 365   1997  do i=888 to 939
  1987  do i=366 to 417   1998  do i=940 to 991
  1988  do i=418 to 470   1999  do i=992 to 1044
  1989  do i=471 to 522   2000  do i=1045 to 1096
  1990  do i=523 to 574   2001  do i=1097 to 1148 */
  1991  do i=575 to 626

array self (i) self1-self9;
do i=1 to 9;
/* start date invalid */
if (stopfl=1 or starfl=1) and self=1 and r4job80>0 then
  do; r4self80=-3; r4job80=-3; end;
if (stopfl=1 or starfl=1) and self=0 and r4job80>0 then
  do; r4emp80=-3; r4job80=-3; end;
if (stopfl=1 or starfl=1) and self=1 and r4job81>0 then
  do; r4self81=-3; r4job81=-3; end;
if (stopfl=1 or starfl=1) and self=0 and r4job81>0 then
  do; r4emp81=-3; r4job81=-3; end;
/*and so on through*/
if (stopfl=1 or starfl=1) and self=1 and r4job01>0 then
  do; r4self01=-3; r4job01=-3; end;
if (stopfl=1 or starfl=1) and self=0 and r4job01>0 then
  do; r4emp01=-3; r4job01=-3; end;
end;

array r4job (i) r4job80-r4job99 r4job00-r4job01;
array r4self (i) r4self80-r4self99 r4self00-r4self01;
array r4emp (i) r4emp80-r4emp99 r4emp00-r4emp01;

*** Include valid skips & non-interview cases;
do over r4job;
  if e200=-5 then do;
    r4job=-5; r4self=-5; r4emp=-5;
  end;
end;

/*merge in data from previous rounds*/
/* R3 variable */ array r3job r3job80-r3job99 r3job00;

```

```

/* Total cv */ array totjob totjob80-totjob99 totjob00;
do over totjob; r3job=totjob; end;

/* (2-26-01) Begin by renaming the infile c.v. variables */
array r1job (i) r1job80-r1job99 r1job00 r1job01;
array r2job (i) r2job80-r2job99 r2job00 r2job01;
array r3job (i) r3job80-r3job99 r3job00 r3job01;
array r4job (i) r4job80-r4job99 r4job00-r4job01;
array r4self (i) r4self80-r4self99 r4self00-r4self01;
array r4emp (i) r4emp80-r4emp99 r4emp00-r4emp01;
array tjob (i) tjob80-tjob99 tjob00 tjob01;
array tsjob (i) tsjob80-tsjob99 tsjob00 tsjob01;
array tejob (i) tejob80-tejob99 tejob00 tejob01;

/* Initialize created variable */
do over tjob;
  tjob=0; tsjob=0; tejob=0;
end;

/* Begin by adding together the created variables from
both rounds. Jobs that are double counted will be
subtracted off later in the program. */
do over tjob;
  if r4job>0 then do; tjob=r4job; end;
  if r4emp>0 then do; tejob=r4emp; end;
  if r4self>0 then do; tsjob=r4self; end;
  if r3job>0 and intwk>0 then do; tjob=r3job;
    tejob=r3job; end;
  if r2job>0 and r3int=-5 and intwk>0 then do;
    tjob=r2job; tejob=r2job; end;
  if r1job>0 and r2int=-5 and r3int=-5 and intwk>0 then
    do; tjob=r1job; tejob=r1job; end;
  if r3job>0 and r4job>0 then do; tjob=r3job+r4job; end;
  if r3job>0 and r4emp>0 then do; tejob=r3job+r4emp;
    end;
  if r2job>0 and r4job>0 and r3int=-5 then do;
    tjob=r2job+r4job; end;
  if r2job>0 and r4emp>0 and r3int=-5 then do;
    tejob=r2job+r4emp; end;
  if r1job>0 and r4job>0 and r2int=-5 and r3int=-5 then
    do; tjob=r1job+r4job; end;
  if r1job>0 and r4emp>0 and r2int=-5 and r3int=-5 then
    do; tejob=r1job+r4emp; end;
end;

/* Define negative values for the total created variable. */
do over tjob;
  if r4job in (-1,-2,-3) then tjob=-3;
  if r4self in (-1,-2,-3) then tsjob=-3;
  if r4emp in (-1,-2,-3) then tejob=-3;
  if r3job in (-1,-2,-3) and intwk>0 then do; tjob=-3;
    tejob=-3; end;
  if r2job in (-1,-2,-3) and r3int=-5 and intwk>0 then do;
    tjob=-3; tejob=-3; end;
  if r1job in (-1,-2,-3) and r2int=-5 and r3int=-5 and
    intwk>0 then do; tjob=-3; tejob=-3; end;
end;

```

```

/* Note there are 7 jobs in Round 1, array matched to
make programming easier */
array r1uid (i) r1uid1-r1uid9;
array r2uid (i) r2uid1-r2uid9;
array r3uid (i) r3uid1-r3uid9;
array r4uid (i) uid1-uid9;

array match1 (i) match11-match19;
array match2 (i) match21-match29;
/*and so on through*/
array match9 (i) match91-match99;

do i=1 to 9;
    match1=0; match2=0; match3=0; match4=0;
    match5=0; match6=0; match7=0; match8=0;
    match9=0;
end;

/* Define match14 as the dummy variable that equals one
when the first job on the Round1 UID roster and the
fourth job on the Round4 roster have the same UID. */

/* If any UID from the first position in a previous round
matches with any UID in Round 4. Need to make sure
that we find the most recent round for that job. */
do over r4uid;
    if r3uid1>0 and r4uid>0 and r3int>0 and intwk>0 then
        do;
            if r3uid1=r4uid then do; match1=3; end;
        end;
    if r2uid1>0 and r4uid>0 and match1=0 and intwk>0
        then do;
            if r2uid1=r4uid then do; match1=2; end;
        end;
    if r1uid1>0 and r4uid>0 and match1=0 and intwk>0
        then do;
            if r1uid1=r4uid then do; match1=1; end;
        end;
    end;

/* If any UID from the second position in a previous
round matches with any UID in Round 4 */
do over r4uid;
do over r4uid;
    if r3uid2>0 and r4uid>0 and r3int>0 and intwk>0 then
        do;
            if r3uid2=r4uid then do; match2=3; end;
        end;
    if r2uid2>0 and r4uid>0 and match2=0 and intwk>0
        then do;
            if r2uid2=r4uid then do; match2=2; end;
        end;
    if r1uid2>0 and r4uid>0 and match2=0 and intwk>0
        then do;
            if r1uid2=r4uid then do; match2=1; end;
        end;
    end;

```

```

/*repeat the above code for UIDs in positions 3-9 in
previous rounds*/

r4match_1=max(match11,match21,match31,match41,
    match51,match61,match71,match81,match91);
r4match_2=max(match12,match22,match32,match42,
    match52,match62,match72,match82,match92);
/*and so on through*/
r4match_9=max(match19,match29,match39,match49,
    match59,match69,match79,match89,match99);

if match11<r4match_1 then match11=0;
if match21<r4match_1 then match21=0;
/*and so on through*/
if match91<r4match_1 then match91=0;

if match12<r4match_2 then match12=0;
if match22<r4match_2 then match22=0;
/*and so on through*/
if match92<r4match_2 then match92=0;

if match13<r4match_3 then match13=0;
if match23<r4match_3 then match23=0;
/*and so on through*/
if match93<r4match_3 then match93=0;

if match14<r4match_4 then match14=0;
if match24<r4match_4 then match24=0;
/*and so on through*/
if match94<r4match_4 then match94=0;

if match15<r4match_5 then match15=0;
if match25<r4match_5 then match25=0;
/*and so on through*/
if match95<r4match_5 then match95=0;

if match16<r4match_6 then match16=0;
if match26<r4match_6 then match26=0;
/*and so on through*/
if match96<r4match_6 then match96=0;

if match17<r4match_7 then match17=0;
if match27<r4match_7 then match27=0;
/*and so on through*/
if match97<r4match_7 then match97=0;

if match18<r4match_8 then match18=0;
if match28<r4match_8 then match28=0;
/*and so on through*/
if match98<r4match_8 then match98=0;

if match19<r4match_9 then match19=0;
if match29<r4match_9 then match29=0;
/*and so on through*/
if match99<r4match_9 then match99=0;

```

Appendix 2: Employment Variable Creation

```

/* Define "same98" as a counter variable that adds up
how many matching UIDs are in the respondent's UID
roster. This will be subtracted from the total number of
jobs created variable to avoid double counting the same
job. Begin by initializing the "same" variables to zero. */
same97=0; same98=0; same99=0; same00=0;

array r1j97 (i) r1j971-r1j979;
array r1j98 (i) r1j981-r1j989;
/*and so on through*/
array r4j01 (i) r4j011-r4j019;

/* Consider the respondents with a Round 1 interview
date in 1997, non-interview cases only.*/
if r1int_y=1997 and intwk>0 then do;
  do over match1;
    if match1=1 and r4j97=1 and r1j971=1 then do;
      same97=same97+1; end;
    end;
  do over match2;
    if match2=1 and r4j97=1 and r1j972=1 then do;
      same97=same97+1; end;
    end;
  /*and so on through*/
  do over match9;
    if match9=1 and r4j97=1 and r1j979=1 then do;
      same97=same97+1; end;
    end;
  end;

/* Consider the respondents with a Round 1 interview
date in 1998, non-interview cases only.*/
if r1int_y=1998 and intwk>0 then do;
  do over match1;
    if match1=1 and r4j98=1 and r1j981=1 then do;
      same98=same98+1; end;
    end;
  /*and so on through*/
  do over match9;
    if match9=1 and r4j98=1 and r1j989=1 then do;
      same98=same98+1; end;
    end;
  end;

/* respondents with a Round 2 interview date in 1998. */
if r2int_y=1998 and intwk>0 then do;
  do over match1;
    if match1=2 and r4j98=1 and r2j981=1 then do;
      same98=same98+1; end;
    end;
  /*and so on through*/
  do over match9;
    if match9=2 and r4j98=1 and r2j989=1 then do;
      same98=same98+1; end;
    end;
  end;

/* respondents with a Round 2 interview date in 1999. */
if r2int_y=1999 and intwk>0 then do;
  do over match1;
    if match1=2 and r4j99=1 and r2j991=1 then do;
      same99=same99+1; end;
    end;
  /*and so on through*/
  do over match9;
    if match9=2 and r4j99=1 and r2j999=1 then do;
      same99=same99+1; end;
    end;
  end;

/* respondents with a Round 3 interview date in 1999. */
if r3int_y=1999 and intwk>0 then do;
  do over match1;
    if match1=3 and r4j99=1 and r3j991=1 then do;
      same99=same99+1; end;
    end;
  /*and so on through*/
  do over match9;
    if match9=3 and r4j99=1 and r3j999=1 then do;
      same99=same99+1; end;
    end;
  end;

/* respondents with a Round 3 interview date in 2000. */
if r3int_y=2000 and intwk>0 then do;
  do over match1;
    if match1=3 and r4j00=1 and r3j001=1 then do;
      same00=same00+1; end;
    end;
  /*and so on through*/
  do over match9;
    if match9=3 and r4j00=1 and r3j009=1 then do;
      same00=same00+1; end;
    end;
  end;

/* Now subtract the "same" count variable from the total
created variable computed above. */
if tjob97=>0 then do; tjob97=tjob97-same97; end;
if tejob97=>0 then do; tejob97=tejob97-same97; end;
if tjob98=>0 then do; tjob98=tjob98-same98; end;
if tejob98=>0 then do; tejob98=tejob98-same98; end;
if tjob99=>0 then do; tjob99=tjob99-same99; end;
if tejob99=>0 then do; tejob99=tejob99-same99; end;
if tjob00=>0 then do; tjob00=tjob00-same00; end;
if tejob00=>0 then do; tejob00=tejob00-same00; end;

/* 12-18-01 Round 4 Non-interview cases */
do over tjob;
  if uid1=-5 then do; tjob=-5; tejob=-5; tsjob=-5; end;
end;

endsas;

```

TOTAL NUMBER OF JOBS HELD AGE 14-19, NUMBER OF EMPLOYEE JOBS HELD SINCE AGE 20, AND NUMBER OF ALL JOBS HELD SINCE AGE 20

Variables Created: CV_TTL_JOBS_TEEN
CV_TTL_JOBS_ADULT_ET
CV_TTL_JOBS_ADULT_ALL

Programs Used

This program uses **emp_begin.sas** as input (see the first page of this appendix for details).

This program calculates the total number of employee-type jobs held by the respondent between the ages of 14 and 19. It then creates two variables for the total number of jobs held by the respondent since age 20: one for all jobs and one for employee-type jobs. See the introduction to this appendix for important information about the universes for each of these variables. The variables are only created for respondents who have worked at least one week in the applicable age range.

```

array job1wks (i) wk1_1-wk1_1148;
array job2wks (i) wk2_1-wk2_1148;
/*and so on through*/
array job9wks (i) wk9_1-wk9_1148;
array starfl (i) starfl_1-starfl_9;
array stopfl (i) stopfl_1-stopfl_9;

** Indicate if worked at least one week on a given job
since age 14 & 20 **
job141=0; job201=0; ejob201=0;
job142=0; job202=0; ejob202=0;
/*and so on through*/
job149=0; job209=0; ejob209=0;

/*Rs under 20 at int date, count jobs worked to date. */
if age14wk>0 and (age20wk-1)>1148 then do;
  do i=age14wk to 1148;
    if job1wks in (-3,1) and self1=0 then job141=1;
    if job2wks in (-3,1) and self2=0 then job142=1;
  /*and so on through*/
    if job9wks in (-3,1) and self9=0 then job149=1;
  end;
end;

/* Rs over 20 at int date, count jobs worked before age
20. */
if age14wk>0 and (age20wk-1)<=1148 then do;
  do i=age14wk to (age20wk-1);
    if job1wks in (-3,1) and self1=0 then job141=1;
    if job2wks in (-3,1) and self2=0 then job142=1;
  /*and so on through*/
    if job9wks in (-3,1) and self9=0 then job149=1;
  end;
end;

/* Rs over 20 at int date, count employee jobs worked
after age 20. */
if age14wk>0 and (age20wk-1)<=1148 then do;
  do i=age20wk to 1148;
    if job1wks in (-3,1) and self1=0 then ejob201=1;

```

```

      if job2wks in (-3,1) and self2=0 then ejob202=1;
      /*and so on through*/
      if job9wks in (-3,1) and self9=0 then ejob209=1;
    end;
  end;

/*Rs over 20 at int date, count jobs worked after age 20.*/
if age14wk>0 and (age20wk-1)<=1148 then do;
  do i=age20wk to 1148;
    if job1wks in (-3,1) then job201=1;
    if job2wks in (-3,1) then job202=1;
  /*and so on through*/
    if job9wks in (-3,1) then job209=1;
  end;
end;

r4job14=sum(job141,job142,job143,job144,job145,
            job146,job147,job148,job149);
r4job20=sum(job201,job202,job203,job204,job205,
            job206,job207,job208,job209);
r4emp20=sum(ejob201,ejob202,ejob203,ejob204,
            ejob205,ejob206,ejob207,ejob208,ejob209);

* only need to assign an invalid skip to cases where the
stop date is unknown and the job starts prior to the
14/20th birthday;
if stopfl_1=1 and starfl_1 ne 1 and starw1<age14wk and
  self1=0 then do; r4job14=-3; end;
if stopfl_1=1 and starfl_1 ne 1 and starw1<age20wk and
  self1=0 and age20wk<intwk then do; r4emp20=-3;
  end;
if stopfl_1=1 and starfl_1 ne 1 and starw1<age20wk and
  age20wk<intwk then do; r4job20=-3; end;
if stopfl_2=1 and starfl_2 ne 1 and starw2<age14wk and
  self2=0 then do; r4job14=-3; end;
if stopfl_2=1 and starfl_2 ne 1 and starw2<age20wk and
  self2=0 and age20wk<intwk then do; r4emp20=-3;
  end;
if stopfl_2=1 and starfl_2 ne 1 and starw2<age20wk and
  age20wk<intwk then do; r4job20=-3; end;

```

Appendix 2: Employment Variable Creation

```

/*and so on through*/
if stopfl_9=1 and starfl_9 ne 1 and starw9<age14wk and
self9=0 then do; r4job14=-3; end;
if stopfl_9=1 and starfl_9 ne 1 and starw9<age20wk and
self9=0 and age20wk<intwk then do; r4emp20=-3;
end;
if stopfl_9=1 and starfl_9 ne 1 and starw9<age20wk and
age20wk<intwk then do; r4job20=-3; end;

if e200=-5 then do;
r4job14=-5; r4job20=-5; r4emp20=-5;
end;

array job14 (i) job141-job149;
array job20 (i) job201-job209;
array r3j14 (i) r3j141-r3j149;
array r3j20 (i) r3j201-r3j209;

do over job14; r3j14=job14; r3j20=job20; end;
r3job14=totjob14; r3job20=totjob20;
drop totjob14 totjob20 job141-job149 job201-job209;

/* Initialize final created variable */
totjob14=0; totjob20=0; totemp20=0;

/* Begin by adding together the created variables from
both rounds. Jobs that are double counted will be
subtracted off later in the program. */
if r4job14>0 then do; totjob14=r4job14; end;
if r3job14>0 and intwk>0 then do; totjob14=r3job14;
end;
if r2job14>0 and r3int=-5 and intwk>0 then do;
totjob14=r2job14; end;
if r1job14>0 and r2int=-5 and r1int=-5 and intwk>0 then
do; totjob14=r1job14; end;

if r3job14>0 and r4job14>0 then do;
totjob14=r3job14+r4job14; end;
if r2job14>0 and r4job14>0 and r3int=-5 then do;
totjob14=r2job14+r4job14; end;
if r1job14>0 and r4job14>0 and r2int=-5 and r3int=-5
then do; totjob14=r1job14+r4job14; end;

if r4job20>0 then do; totjob20=r4job20; end;
if r3job20>0 and intwk>0 then do; totjob20=r3job20;
end;
if r3job20>0 and r4job20>0 then do;
totjob20=r3job20+r4job20; end;

if r4emp20>0 then do; totemp20=r4emp20; end;
if r3job20>0 and intwk>0 then do; totemp20=r3job20;
end;
if r3job20>0 and r4emp20>0 then do;
totemp20=r3job20+r4emp20; end;

/* Jobs that are worked in 1998 or 1999 have the
possibility of being double counted since they are

```

counted in each Round's created variable programs. Here we will try to compare the UID's to see which jobs are double counted. */

```

array r3j14 (i) r3j141-r3j149;
array r3j20 (i) r3j201-r3j209;
array job14 (i) job141-job149;
array job20 (i) job201-job209;
array r1uid (i) r1uid1-r1uid7;
array r2uid (i) r2uid1-r2uid9;
array r3uid (i) r3uid1-r3uid9;
array r4uid (i) uid1-uid9;
array match1 (i) match11-match19;
array match2 (i) match21-match29;
array match3 (i) match31-match39;
array match4 (i) match41-match49;
array match5 (i) match51-match59;
array match6 (i) match61-match69;
array match7 (i) match71-match79;
array match8 (i) match81-match89;
array match9 (i) match91-match99;

do i=1 to 9;
match1=0; match2=0; match3=0; match4=0;
match5=0; match6=0; match7=0; match8=0;
match9=0;
end;

/* Define match14 as the dummy variable that equals one
when the first job on the R1 UID roster and the
fourth job on the R4 roster have the same UID. */

/* If any UID from the first position in dli Round
matches with any UID in Round 4 */
do over r4uid;
if r3uid1>0 and r4uid>0 and r3int>0 and intwk>0 then
do;
if r3uid1=r4uid then do; match1=1; end;
end;
if r2uid1>0 and r4uid>0 and r3int=-5 and intwk>0
then do;
if r2uid1=r4uid then do; match1=1; end;
end;
if r1uid1>0 and r4uid>0 and r3int=-5 and r2int=-5 and
intwk>0 then do;
if r1uid1=r4uid then do; match1=1; end;
end;
if r1uid1>0 and r4uid>0 and r3int=-5 and r2int=-5 and
intwk>0 then do;
if r1uid1=r4uid then do; match1=1; end;
end;

/* If any UID from the second position in dli Round
matches with any UID in Round 4 */
do over r4uid;
if r3uid2>0 and r4uid>0 and r3int>0 and intwk>0 then
do;
if r3uid2=r4uid then do; match2=1; end;
end;
if r2uid2>0 and r4uid>0 and r3int=-5 and intwk>0
then do;

```

```

        if r2uid2=r4uid then do; match2=1; end;
end;
if r1uid2>0 and r4uid>0 and r3int=-5 and r2int=-5 and
    intwk>0 then do;
    if r1uid2=r4uid then do; match2=1; end;
    end;
end;

/*The above code repeats for UIDs in positions 3
through 9 in the last interview round. Contact NLS User
Services for more information.*/

/* Define "same" as a counted variable that adds up how
many matching UIDs are in the respondent's UID
roster. This will be subtracted from the total
number of jobs created variable to avoid double
counting the same job. Begin by initializing the
"same" variables to zero.*/
same14=0;
same20=0;

do over match1;
if match1=1 and r3j141=1 and job14=1 then do;
    same14=same14+1; end;
if match1=1 and job14=1 and r3int=-5 then do;
    same14=same14+1; end;
if match1=1 and r3j201=1 and job20=1 then do;
    same20=same20+1; end;
end;

do over match2;
if match2=1 and r3j142=1 and job14=1 then do;
    same14=same14+1; end;
if match2=1 and job14=1 and r3int=-5 then do;
    same14=same14+1; end;
if match2=1 and r3j202=1 and job20=1 then do;
    same20=same20+1; end;

```

```

end;
/*repeat the above for match3 through match9*/

/* Now subtract the "same" count variable from the total
created variable computed above. */
if totjob14=>0 then do; totjob14=totjob14-same14; end;
if totjob20=>0 then do; totjob20=totjob20-same20; end;
if totemp20=>0 then do; totemp20=totemp20-same20;
end;

/* Define negative values for the total created variable.*/
if r1job14 in (-1,-2,-3) and intwk>0 and r3int=-5 and
    r2int=-5 then totjob14=-3;
if r2job14 in (-1,-2,-3) and intwk>0 and r3int=-5 then
    totjob14=-3;
if r3job14 in (-1,-2,-3) and intwk>0 then totjob14=-3;
if r4job14 in (-1,-2,-3) then totjob14=-3;

if r3job20 in (-1,-2,-3) and intwk>0 then totjob20=-3;
if r4job20 in (-1,-2,-3) then totjob20=-3;

if r3job20 in (-1,-2,-3) and intwk>0 then totemp20=-3;
if r4emp20 in (-1,-2,-3) then totemp20=-3;

/* For respondents younger than 20 at interview date,
   their thrs14 c.v. will be -4. */
if age20wks=intwk and intwk>0 then do;
    totjob20=-4; totemp20=-4;
end;

if uid1=-5 then do;
    totjob14=-5; totjob20=-5; totemp20=-5;
end;

endsas;

```

Appendix 2: Employment Variable Creation

HOURLY RATE OF PAY, HOURLY MONETARY COMPENSATION, AND JOB LENGTH < 13 WEEKS

Variables Created:

- CV_HRLY_PAY
- CV_HRLY_COMPENSATION
- CV_JOB<13_WKS

Variables Used

Name in Program	Question Name on CD	Name in Program	Question Name on CD
e204	YEMP-204	e2260411-e2260419	YEMP-22604.01~000001~000009
e6001-e6007	YEMP-600.01-.07	e2260410	YEMP-22604.01~000010
e192001-e192009	YEMP-19200.01-.09	e2260421-e2260429	YEMP-22604.02~000001~000009
e207001-e207009	YEMP-20700.01-.09	e2260420	YEMP-22604.02~000010
e230001-e230009	YEMP-23000.01-.09	e2260431-e2260439	YEMP-22604.03~000001~000009
e232001-e232003, e232007	YEMP-23200.01-.03, .07	e2260430	YEMP-22604.03~000010
e239001, e239002, e239005	YEMP-23900.01, .02, .05	e2260441-e2260449	YEMP-22604.04~000001~000009
e239011-e239019	YEMP-23901.01-.09	e2260440	YEMP-22604.04~000010
e245011-e245016, e245018	YEMP-24501.01-.06, .08	e2260451-e2260459	YEMP-22604.05~000001~000009
e245021-e245026, e245028	YEMP-24502.01-.06, .08	e2260450	YEMP-22604.05~000010
e245141-e245146	YEMP-24514.01-.06	e2260461-e2260469	YEMP-22604.06~000001~000009
e24514b1	YEMP-24514B.01	e2260460	YEMP-22604.06~000010
e2451711, e2451712	YEMP-24517.01.01, .02	e2260471-e2260479	YEMP-22604.07~000001~000009
e2680011, e2680012	YEMP-26800.01.01, .02	e2120011-e2120016	YEMP-21200.01~000001~000006
e334001-e334007	YEMP-33400.01-.07	e2120021-e2120026	YEMP-21200.02~000001~000006
e335001-e335005	YEMP-33500.01-.05	e2120031-e2120036	YEMP-21200.03~000001~000006
e336001-e336005	YEMP-33600.01-.05	e2120041-e2120046	YEMP-21200.04~000001~000006
e344021-e344024	YEMP-34402.01-.07	e2120051-e2120056	YEMP-21200.05~000001~000006
e34402aab1-e34402aab5	YEMP-34402AAB.01-.05	e2120061-e2120066	YEMP-21200.06~000001~000006
e34402aad1-e34402aad5	YEMP-34402AAD.01-.05	e2120071-e2120076	YEMP-21200.07~000001~000006
e34402ab1-e34402ab5	YEMP-34402AB.01-.05	e2120081-e2120086	YEMP-21200.08~000001~000006
e34403a1-e34403a3	YEMP-34403.01-.03	e382111-e382115	YEMP-38211.01-.05
e34404a1, e34404a2	YEMP-34404.01, .02	e38329b1-e38329b4	YEMP-38329B.01-.04
e3440403	YEMP-34404.03	e38329d1-e38329d4	YEMP-38329D.01-.04
e34413c1-e34413c3	YEMP-34413C.01-.03	e38329i1, e38329i4	YEMP-38329I.01, .04
e34413e1	YEMP-34413E.01	e38329i1b, e38329i4b	YEMP-38329IB.01, .04
e3441611, e3441612	YEMP-34416.01.01, .02	e38329k1-e38329k3	YEMP-38329K.01, .03
e3442511, e3442512	YEMP-34425.01.01, .02	e3840711-e3840715	YEMP-38407.01.01-.05
e3442611, e3442612	YEMP-34426.01.01, .02	e3840721-e3840725	YEMP-38407.02.01-.05
e344281-e344287	YEMP-34428.01-.07	e3840731-e3840734	YEMP-38407.03.01-.04
e344291-e344295	YEMP-34429.01-.05	e3840741-e3840745	YEMP-38407.04.01-.05
e344301	YEMP-34430.01	e3840751	YEMP-38407.05.01
e356001-e356004	YEMP-35600.01-.04	e3841611-e3841615	YEMP-38416.01.01-.05
e361001-e361004	YEMP-36100.01-.04	e3841621-e3841625	YEMP-38416.02.01-.05
e362001-e362004	YEMP-36200.01-.04	e3841631-e3841634	YEMP-38416.03.01-.04
e2160012-e2160016	YEMP-21600.01.02-.06	e3841641-e3841643, e3841645	YEMP-38416.04.01-.03, .05
e2160022-e2160026	YEMP-21600.02.02-.06	e3841651	YEMP-38416.05.01
e2160032-e2160036	YEMP-21600.03.02-.06	e3841611-e3841615	YEMP-38416I.01.01-.05
e2160042-e2160046	YEMP-21600.04.02-.06	e3841622, e3841624	YEMP-38416I.02.02, .04
e2160052-e2160055	YEMP-21600.05.02-.05	e38416ib11-e38416ib135	YEMP-38416IB.01.01-.05
e2160062-e2160065	YEMP-21600.06.02-.05	e38416ib22, e38416ib24	YEMP-38416IB.02.02, .04
e2160072, e2160073	YEMP-21600.07.02, .03	e38416k11-e38416k15	YEMP-38416K.01.01-.05
e2160083, e2160084	YEMP-21600.08.03, .04	e38416k21, e38416k25	YEMP-38416K.02.01, .05
e2250012-e2250016	YEMP-22500.01.02-.06	e38416k31	YEMP-38416K.03.01
e2250022-e2250026	YEMP-22500.02.02-.06	e38416k45	YEMP-38416K.04.05
e2250032-e2250036	YEMP-22500.03.02-.06	e38416k51	YEMP-38416K.05.01
e2250042-e2250046	YEMP-22500.04.02-.06	e582011-e582019	YEMP-58201.01-.09
e2250052-e2250055	YEMP-22500.05.02-.05	e599001-e599009	YEMP-59900.01-.09
e2250062-e2250065	YEMP-22500.06.02-.05	e831001-e831009	YEMP-83100.01-.09
e2250072, e2250073	YEMP-22500.07.02, .03	e868001-e868007	YEMP-86800.01-.07
e2250083, e2250084	YEMP-22500.08.03, .04	e869001-e869009	YEMP-86900.01-.09
e2253012-e2253015	YEMP-22530.01.02-.05	e871001-e871004	YEMP-87100.01-.04
e2253022-e2253026	YEMP-22530.02.02-.06	e880001-e880009	YEMP-88000.01-.09

Appendix 2: Employment Variable Creation

e2253032-e2253034	YEMP-22530.03.02-.04	e885011-e885017	YEMP-88501.01-.07
e2253042-e2253046	YEMP-22530.04.02-.06	e885021-e885027	YEMP-88502.01-.07
e2253512-e2253515	YEMP-22535.01.02-.05	e885121-e885127	YEMP-88512.01-.07
e2253522-e2253526	YEMP-22535.02.02-.06	e973001-e973005	YEMP-97300.01-.05
e2253532-e2253534	YEMP-22535.03.02-.04	e974001-e974007	YEMP-97400.01-.07
e2253542-e2253546	YEMP-22535.04.02-.06	e975001-e975004	YEMP-97500.01-.04
e2255012-e2255016	YEMP-22550.01.02-.06	e982001-e982003	YEMP-98200.01-.03
e2255022-e2255026	YEMP-22550.02.02-.06	e982005, e982007	YEMP-98200.05, .07
e2255032-e2255035	YEMP-22550.03.02-.05	e983001-e983005	YEMP-98300.01-.05
e2255042	YEMP-22550.04.02	e984021-e984027	YEMP-98402.01-.07
e2255052	YEMP-22550.05.02	e98402caa1-e98402caa5	YEMP-98402CAA.01-.05
e2255062, e2255065	YEMP-22550.06.02, .05	e98402cab1-e98402cab5	YEMP-98402CAB.01-.05
e2255072	YEMP-22550.07.02	e98402d1-e98402d5	YEMP-98402D.01-.05
e2260470	YEMP-22604.07-000010	e984031-e984033	YEMP-98403.01-.03
e226091-e226095	YEMP-22609.01.01-.05.01	e98404a1-e98404a3	YEMP-98404A.01-.03
e226101-e226103	YEMP-22610.01.02-.03.02	e984141-e984143	YEMP-98414.01-.03
e226111-e226113	YEMP-22611.01.03-.03.03	e98414aaa1, e98414aaa2	YEMP-98414AAA.01, .02
e226121-e226123	YEMP-22612.01.04-.03.04	e98414aab1, e98414aab2	YEMP-98414AAB.01, .02
e226131-e226133	YEMP-22613.01.05-.03.05	e98414b2, e98414b3	YEMP-98414B.02, .03
e226141, e226142	YEMP-22614.01.06, .02.06	e984291-e984293	YEMP-98429.01-.03
e226151	YEMP-22615.01.07	e995001-e995005	YEMP-99500.01-.05
e226161, e226163	YEMP-22616.01.08, .03.08	e1000001-e1000005	YEMP-100000.01-.05
e226171	YEMP-22617.01.09	e1001001-e1001004	YEMP-100100.01-.04
e226261-e226265	YEMP-22626.01.01-.05.01	e1020512-e1020516	YEMP-100205.01.02-.06
e22626c2	YEMP-22626C.02.01	e1020522-e1020526	YEMP-100205.02.02-.06
e22626cb2	YEMP-22626CB.02.01	e1020532-e1020536	YEMP-100205.03.02-.06
e22626e1	YEMP-22626E.01.01	e1020542-e1020546	YEMP-100205.04.02-.06
e226271-e226273	YEMP-22627.01.04-.03.04	e1020552-e1020554	YEMP-100205.05.02-.04
e226281-e226283	YEMP-22628.01.05-.03.05	e1020572	YEMP-100205.07.02
e22628c1	YEMP-22628C.01.05	e1021412-e1021416	YEMP-100214.01.02-.06
e22628cb1	YEMP-22628CB.01.05	e1021422-e1021426	YEMP-100214.02.02-.06
e22628e1-e22628e3	YEMP-22628E.01.05-.03.05	e1021432-e1021436	YEMP-100214.03.02-.06
e226291, e226292	YEMP-22629.01.06, .02.06	e1021442-e1021446	YEMP-100214.04.02-.06
e22629c2	YEMP-22629C.02.06	e1021452-e1021454	YEMP-100214.05.02-.04
e22629cb2	YEMP-22629CB.02.06	e1021472	YEMP-100214.07.02
e226301	YEMP-22630.01.07	e10214aaa12-e10214aaa15	YEMP-100214AAA.01.02-.05
e22630e1	YEMP-22630E.01.07	e10214aaa24, e10214aaa26	YEMP-100214AAA.02.04, .06
e226311, e226313	YEMP-22631.01.08, .03.08	e10214aaa32, e10214aaa36	YEMP-100214AAA.03.02, .06
e22631c1	YEMP-22631C.01.08	e10214aaa53	YEMP-100214AAA.05.03
e22631cb1	YEMP-22631CB.01.08	e10214aab12-e10214aab15	YEMP-100214AAB.01.02-.05
e226321	YEMP-22632.01.09	e10214aab24, e10214aab26	YEMP-100214AAB.02.04, .06
e377011-e377019	YEMP-37701.01.09	e10214aab32, e10214aab36	YEMP-100214AAB.03.02, .06
e37901b1-e37901b9	YEMP-37901B.01-.09	e10214aab53	YEMP-100214AAB.05.03
e3800b1-e3800b7	YEMP-3800B.01-.07	e10214b12-e10214b15	YEMP-100214B.01.02-.05
e3800b9	YEMP-3800B.09	e10214b22-e10214b24	YEMP-100214B.02.02-.04
e3800f1-e3800f7	YEMP-3800F.01-.07	e10214b32	YEMP-100214B.03.02
e380011-e380017	YEMP-38001.01-.07	e10214b42	YEMP-100214B.04.02
e380019	YEMP-38001.09	e10214b52	YEMP-100214B.05.02
e380021-e380024	YEMP-38002.01-.04	e10214b72	YEMP-100214B.07.02
e380026	YEMP-38002.06	e1022511-e1022519	YEMP-100225.01-000001-~000009
e380031-e380034	YEMP-38003.01-.04	e1022510	YEMP-100225.01-000010
e380036	YEMP-38003.06	e1022521-e1022529	YEMP-100225.02-000001-~000009
e380121-e380124	YEMP-38012.01-.04	e1022520	YEMP-100225.02-000010
e380126	YEMP-38012.06	e1022531-e1022539	YEMP-100225.03-000001-~000009
e38012b1, e38012b2	YEMP-38012B.01, .02	e1022530	YEMP-100225.03-000010
e380131-e380137	YEMP-38013.01-.07	e1022541-e1022549	YEMP-100225.04-000001-~000009
e380139	YEMP-38013.09	e1022540	YEMP-100225.04.000010
e38013b1- e38013b3	YEMP-38013B.01-.03	e102301-e102304	YEMP-100230.01.01-.04.01
e380141-e380145	YEMP-38014.01-.05	e102311-e102314	YEMP-100231.01.02-.04.02
e380231-e380235	YEMP-38023.01-.05	e102321-e102323	YEMP-100232.01.03-.03.03
e38024b1-e38024b3	YEMP-38024B.01-.03	e102331, e102332	YEMP-100233.01.04, .02.04
e38024c1-e38024c3	YEMP-38024C.01-.03	e102341, e102342	YEMP-100234.01.05, .02.05

Appendix 2: Employment Variable Creation

e380271, e380272	YEMP-38027.01, .02	e102351	YEMP-100235.01.06
e381011-e381016	YEMP-38101.01-.06	e102362	YEMP-100236.02.07
e381021-e381025	YEMP-38102.01-.05	e102371	YEMP-100237.01.08
e381031-e381033	YEMP-38103.01-.03	e102481-e102484	YEMP-100248.01.01-.04.01
e381051-e381055	YEMP-38105.01-.05	e10248e1	YEMP-100248E.01.01
e381061-e381066	YEMP-38106.01-.06	e102491, e102492	YEMP-100249.01.04, .02.04
e381071-e381075	YEMP-38107.01-.05	e102501, e102502	YEMP-100250.01.05, .02.05
e229001-e229008	YEMP-22900.01-.08	e102511	YEMP-100251.01.06
e381161-e381165	YEMP-38116.01-.05	e102522	YEMP-100252.02.07
e38116aaa1	YEMP-38116AAA.01	e10252e2	YEMP-100252E.02.07
e38116aab1	YEMP-38116AAB.01	e102531	YEMP-100253.01.08
e38116b1, e38116b2	YEMP-38116B.01, .02	id	PUBID
e382011-e382015	YEMP-38201.01-.05	e383291- e383296	
e382021-e382025	YEMP-38202.01-.05		

Codes for Created Variable

Note that hourly rate of pay is reported with two implied decimal places.

This program creates the hourly rate of pay for NLSY97 respondents. The hourly rate of pay is constructed from stop date information for respondents whose job lasted more than 13 weeks. For all other respondents the start wage is used.

In addition, this program creates an hourly monetary compensation variable for NLSY97 respondents. This variable that includes information about all compensation received by the respondent, such as tips, bonuses, commissions, overtime, etc., in the calculation. Hourly monetary compensation differs from hourly rate of pay variable, which calculates only the base pay rate. This variable is constructed from stop date information for respondents with jobs longer than 13 weeks and start date information for other.

Finally, a variable indicating whether the jobs lasted more than 13 weeks is also created. There are up to 9 jobs reported, so each variable is created for 9 jobs.

```
/* DECLARING THE ARRAYS TO BE USED LATER IN THE PROGRAM*/
array e19200 e192001-e192009; array e37901b e37901b1-e37901b9; array e59900 e599001-e599009;
array e22900 e229001-e229009; array e23000 e230001-e230009; array e23200 e232001-e232009;
array e23900 e239001-e239009; array e33400 e334001-e334009; array e34402 e344021-e344029;
array e34402ab e34402ab1-e34402ab9; array e33600 e336001-e336009; array e33500 e335001-e335009;
array e34428 e344281-e344289; array e34430 e344301-e344309; array e36100 e361001-e361009;
array e36200 e362001-e362009; array e20700 e207001-e207009; array e24501 e245011-e245019;
array e24514 e245141-e245149; array e24514b e24514b1-e24514b9; array e38013 e380131-e380139;
array e38014 e380141-e380149; array e38023 e380231-e380239; array e38106 e381061-e381069;
array e38107 e381071-e381079; array e38116 e381161-e381169; array e3800b e3800b1-e3800b9;
array e38027 e380271-e380279; array e23901 e239011-e239019; array e3800f e3800f1-e3800f9;
array e38101 e381011-e381019; array e38102 e381021-e381029; array e38103 e381031-e381039;
array e38201 e382011-e382019; array e38105 e381051-e381059; array e24502 e245021-e245029;
array e38202 e382021-e382029; array e38211 e382111-e382119; array e38211b e38211b1-e38211b9;
array e34400 e344001-e344009; array e38313 e383131-e383139; array e83100 e831001-e831009;
array e86800 e868001-e868009; array e86900 e869001-e869009; array e87100 e871001-e871009;
array e87800 e878001-e878009; array e88000 e880001-e880009; array e97300 e973001-e973009;
array e97400 e974001-e974009; array e97500 e975001-e975009; array e98300 e983001-e983009;
array e98402 e984021-e984029; array e98402d e98402d1-e98402d9; array e98429 e984291-e984299;
array e98429l e98429l1-e98429l9; array e100000 e1000001-e1000009; array e100100 e1001001-e1001009;
array e99500 e995001-e995009; array e88501 e885011-e885019; array e88502 e885021-e885029;
array e88512 e885121-e885129; array e88512b e88512b1-e88512b9; array e34403 e344031-e344039;
array e34404 e344041-e344049; array e34413c e34413c1-e34413c9; array e34413e e34413e1-e34413e9;
array e98403 e984031-e984039; array e98404a e98404a1-e98404a9; array e98414 e984141-e984149;
array e98414b e98414b1-e98414b9; array e38001 e380011-e380019; array e38002 e380021-e380029;
array e38003 e380031-e380039; array e38012 e380121-e380129; array e38012b e38012b1-e38012b9;
array e38329b e38329b1-e38329b9; array e38329d e38329d1-e38329d9; array e38329 e383291-e383299;
```

```

array e38330 e383301-e383309;           array e35600 e356001-e356009;           array e22609 e226091-e226099;
array e22610 e226101-e226109;          array e22611 e226111-e226119;          array e22612 e226121-e226129;
array e22613 e226131-e226139;          array e22614 e226141-e226149;          array e22615 e226151-e226159;
array e22616 e226161-e226269;          array e22617 e226171-e226179;          array e22626 e226261-e226269;
array e22627 e226271-e226279;          array e22628 e226281-e226289;          array e22629 e226291-e226299;
array e22630 e226301-e226309;          array e22631 e226311-e226319;          array e22632 e226321-e226329;
array e100230 e102301-e102309;          array e100231 e102311-e102319;          array e100232 e102321-e102329;
array e100233 e102331-e102339;          array e100234 e102341-e102349;          array e100235 e102351-e102359;
array e100236 e102361-e102369;          array e100237 e102371-e102379;          array e100239 e102391-e102399;
array e100248 e102481-e102489;          array e100249 e102491-e102499;          array e100250 e102501-e102509;
array e100251 e102511-e102519;          array e100252 e102521-e102529;          array e100253 e102531-e102539;
array e100254 e102541-e102549;          array e37701 e377011-e377019;          array e58401 e584011-e584019;
array e58201 e582011-e582019;          array euid euid1-euid9;                array e34402aab e34402aab1-e34402aab9;
array e38116b e38116b1-e38116b9;        array pay pay01-pay09;                array e98402cab e98402cab1-e98402cab9;
array e34402aad e34402aad1-e34402aad9; array e22626cb e22626cb1-e22626cb9;    array e100248c e100248c1-e100248c9;
array e98402caa e98402caa1-e98402caa9; array e22626e e22626e1-e22626e9;       array e100248e e100248e1-e100248e9;
array e22626c e22626c1-e22626c9;        array e24514aab e24514aab1-e24514aab9;   array e24514aabb e24514aabb1-e24514aabb9;
array e22626e e22626e1-e22626e9;        array e34413cd e34413cd1-e34413cd9;    array e34413cd e34413cd1-e34413cd9;
array e100248cb e100248cb1-e100248cb9; array e88512aab e88512aab1-e88512aab9;  array e88512aab e88512aab1-e88512aab9;
array e24514aab e24514aab1-e24514aab9; array e98414aab e98414aab1-e98414aab9;  array e98414aab e98414aab1-e98414aab9;
array e34413cc e34413cc1-e34413cc9;     array e22627cb e22627cb1-e22627cb9;    array e22627cb e22627cb1-e22627cb9;
array e88512aaa e88512aaa1-e88512aaa9; array e100249c e100249c1-e100249c9;    array e100249c e100249c1-e100249c9;
array e98414aaa e98414aaa1-e98414aaa9; array e100249e e100249e1-e100249e9;    array e100249e e100249e1-e100249e9;
array e22627c e22627c1-e22627c9;        array e22628cb e22628cb1-e22628cb9;    array e22628cb e22628cb1-e22628cb9;
array e22627e e22627e1-e22627e9;        array e100250c e100250c1-e100250c9;   array e100250c e100250c1-e100250c9;
array e100249cb e100249cb1-e100249cb9; array e100250e e100250e1-e100250e9;   array e100250e e100250e1-e100250e9;
array e22628c e22628c1-e22628c9;        array e22629cb e22629cb1-e22629cb9;   array e22629cb e22629cb1-e22629cb9;
array e22628e e22628e1-e22628e9;        array e100251c e100251c1-e100251c9;   array e100251c e100251c1-e100251c9;
array e100250cb e100250cb1-e100250cb9; array e100251e e100251e1-e100251e9;   array e100251e e100251e1-e100251e9;
array e22629c e22629c1-e22629c9;        array e22630cb e22630cb1-e22630cb9;   array e22630cb e22630cb1-e22630cb9;
array e22629e e22629e1-e22629e9;        array e100252c e100252c1-e100252c9;   array e100252c e100252c1-e100252c9;
array e100251cb e100251cb1-e100251cb9; array e100252e e100252e1-e100252e9;   array e100252e e100252e1-e100252e9;
array e22630c e22630c1-e22630c9;        array e22631cb e22631cb1-e22631cb9;  array e22631cb e22631cb1-e22631cb9;
array e22630e e22630e1-e22630e9;        array e100253c e100253c1-e100253c9;  array e100253c e100253c1-e100253c9;
array e100252cb e100252cb1-e100252cb9; array e100253e e100253e1-e100253e9;  array e100253e e100253e1-e100253e9;
array e22631c e22631c1-e22631c9;        array e22632cb e22632cb1-e22632cb9;  array e22632cb e22632cb1-e22632cb9;
array e22631e e22631e1-e22631e9;        array e100254c e100254c1-e100254c9;  array e100254c e100254c1-e100254c9;
array e100253cb e100253cb1-e100253cb9; array e100254e e100254e1-e100254e9;  array e100254e e100254e1-e100254e9;
array e22632c e22632c1-e22632c9;        array e38024c e38024c1-e38024c9;    array e38024c e38024c1-e38024c9;
array e22632e e22632e1-e22632e9;        array endhr1 endhr11-endhr19;      array endhr2 endhr21-endhr29;
array e100254cb e100254cb1-e100254cb9; array e38116aaa e38116aaa1-e38116aaa9; array e38116aab e38116aab1-e38116aab9;
array e38012aaa e38012aaa1-e38012aaa9; array e38012aab e38012aab1-e38012aab9; array e38211aab e38211aab1-e38211aab9;
array e38211aaa e38211aaa1-e38211aaa9; array e38329g e38329g1-e38329g9;      array e38329i e38329i1-e38329i9;
array e38329ib e38329ib1-e38329ib9;      array e38329k e38329k1-e38329k9;
array e225001 e2250012 e2250022 e2250032 e2250042 e2250052 e2250062 e2250072 e2250082 e2250092;
array e225002 e2250013 e2250023 e2250033 e2250043 e2250053 e2250063 e2250073 e2250083 e2250093;
array e225003 e2250014 e2250024 e2250034 e2250044 e2250054 e2250064 e2250074 e2250084 e2250094;
array e225004 e2250015 e2250025 e2250035 e2250045 e2250055 e2250065 e2250075 e2250085 e2250095;
array e225005 e2250016 e2250026 e2250036 e2250046 e2250056 e2250066 e2250076 e2250086 e2250096;
array e225301 e2253012 e2253022 e2253032 e2253042 e2253052 e2253062 e2253072 e2253082 e2253092;
array e225302 e2253013 e2253023 e2253033 e2253043 e2253053 e2253063 e2253073 e2253083 e2253093;

```

Appendix 2: Employment Variable Creation

array e225303 e2253014 e2253024 e2253034 e2253044 e2253054 e2253064 e2253074 e2253084 e2253094;
array e225304 e2253015 e2253025 e2253035 e2253045 e2253055 e2253065 e2253075 e2253085 e2253095;
array e225305 e2253016 e2253026 e2253036 e2253046 e2253056 e2253066 e2253076 e2253086 e2253096;

array e225351 e2253512 e2253522 e2253532 e2253542 e2253552 e2253562 e2253572 e2253582 e2253592;
array e225352 e2253513 e2253523 e2253533 e2253543 e2253553 e2253563 e2253573 e2253583 e2253593;
array e225353 e2253514 e2253524 e2253534 e2253544 e2253554 e2253564 e2253574 e2253584 e2253594;
array e225354 e2253515 e2253525 e2253535 e2253545 e2253555 e2253565 e2253575 e2253585 e2253595;
array e225355 e2253516 e2253526 e2253536 e2253546 e2253556 e2253566 e2253576 e2253586 e2253596;

array e225501 e2255012 e2255022 e2255032 e2255042 e2255052 e2255062 e2255072 e2255082 e2255092;
array e225502 e2255013 e2255023 e2255033 e2255043 e2255053 e2255063 e2255073 e2255083 e2255093;
array e225503 e2255014 e2255024 e2255034 e2255044 e2255054 e2255064 e2255074 e2255084 e2255094;
array e225504 e2255015 e2255025 e2255035 e2255045 e2255055 e2255065 e2255075 e2255085 e2255095;
array e225505 e2255016 e2255026 e2255036 e2255046 e2255056 e2255066 e2255076 e2255086 e2255096;

array e216001 e2160012 e2160022 e2160032 e2160042 e2160052 e2160062 e2160072 e2160082 e2160092;
array e216002 e2160013 e2160023 e2160033 e2160043 e2160053 e2160063 e2160073 e2160083 e2160093;
array e216003 e2160014 e2160024 e2160034 e2160044 e2160054 e2160064 e2160074 e2160084 e2160094;
array e216004 e2160015 e2160025 e2160035 e2160045 e2160055 e2160065 e2160075 e2160085 e2160095;
array e216005 e2160016 e2160026 e2160036 e2160046 e2160056 e2160066 e2160076 e2160086 e2160096;

array e212001 e2120011 e2120021 e2120031 e2120041 e2120051 e2120061 e2120071 e2120081 e2120091;
array e212002 e2120012 e2120022 e2120032 e2120042 e2120052 e2120062 e2120072 e2120082 e2120092;
array e212003 e2120013 e2120023 e2120033 e2120043 e2120053 e2120063 e2120073 e2120083 e2120093;
array e212004 e2120014 e2120024 e2120034 e2120044 e2120054 e2120064 e2120074 e2120084 e2120094;
array e212005 e2120015 e2120025 e2120035 e2120045 e2120055 e2120065 e2120075 e2120085 e2120095;
array e212006 e2120016 e2120026 e2120036 e2120046 e2120056 e2120066 e2120076 e2120086 e2120096;

array e384161 e3841611 e3841621 e3841631 e3841641 e3841651 e3841661 e3841671 e3841681 e3841691;
array e384162 e3841612 e3841622 e3841632 e3841642 e3841652 e3841662 e3841672 e3841682 e3841692;
array e384163 e3841613 e3841623 e3841633 e3841643 e3841653 e3841663 e3841673 e3841683 e3841693;
array e384164 e3841614 e3841624 e3841634 e3841644 e3841654 e3841664 e3841674 e3841684 e3841694;
array e384165 e3841615 e3841625 e3841635 e3841645 e3841655 e3841665 e3841675 e3841685 e3841695;

array e38416g1 e38416g11 e38416g21 e38416g31 e38416g41 e38416g51 e38416g61 e38416g71 e38416g81 e38416g91;
array e38416g2 e38416g12 e38416g22 e38416g32 e38416g42 e38416g52 e38416g62 e38416g72 e38416g82 e38416g92;
array e38416g3 e38416g13 e38416g23 e38416g33 e38416g43 e38416g53 e38416g63 e38416g73 e38416g83 e38416g93;
array e38416g4 e38416g14 e38416g24 e38416g34 e38416g44 e38416g54 e38416g64 e38416g74 e38416g84 e38416g94;
array e38416g5 e38416g15 e38416g25 e38416g35 e38416g45 e38416g55 e38416g65 e38416g75 e38416g85 e38416g95;

array e38416i1 e38416i11 e38416i21 e38416i31 e38416i41 e38416i51 e38416i61 e38416i71 e38416i81 e38416i91;
array e38416i2 e38416i12 e38416i22 e38416i32 e38416i42 e38416i52 e38416i62 e38416i72 e38416i82 e38416i92;
array e38416i3 e38416i13 e38416i23 e38416i33 e38416i43 e38416i53 e38416i63 e38416i73 e38416i83 e38416i93;
array e38416i4 e38416i14 e38416i24 e38416i34 e38416i44 e38416i54 e38416i64 e38416i74 e38416i84 e38416i94;
array e38416i5 e38416i15 e38416i25 e38416i35 e38416i45 e38416i55 e38416i65 e38416i75 e38416i85 e38416i95;

array e38416ib1 e38416ib11 e38416ib21 e38416ib31 e38416ib41 e38416ib51 e38416ib61 e38416ib71 e38416ib81
e38416ib91;
array e38416ib2 e38416ib12 e38416ib22 e38416ib32 e38416ib42 e38416ib52 e38416ib62 e38416ib72 e38416ib82
e38416ib92;
array e38416ib3 e38416ib13 e38416ib23 e38416ib33 e38416ib43 e38416ib53 e38416ib63 e38416ib73 e38416ib83
e38416ib93;
array e38416ib4 e38416ib14 e38416ib24 e38416ib34 e38416ib44 e38416ib54 e38416ib64 e38416ib74 e38416ib84
e38416ib94;
array e38416ib5 e38416ib15 e38416ib25 e38416ib35 e38416ib45 e38416ib55 e38416ib65 e38416ib75 e38416ib85
e38416ib95;

```

array e38416k1 e38416k11 e38416k21 e38416k31 e38416k41 e38416k51 e38416k61 e38416k71 e38416k81
e38416k91;
array e38416k2 e38416k12 e38416k22 e38416k32 e38416k42 e38416k52 e38416k62 e38416k72 e38416k82
e38416k92;
array e38416k3 e38416k13 e38416k23 e38416k33 e38416k43 e38416k53 e38416k63 e38416k73 e38416k83
e38416k93;
array e38416k4 e38416k14 e38416k24 e38416k34 e38416k44 e38416k54 e38416k64 e38416k74 e38416k84
e38416k94;
array e38416k5 e38416k15 e38416k25 e38416k35 e38416k45 e38416k55 e38416k65 e38416k75 e38416k85
e38416k95;

array e384071 e3840711 e3840721 e3840731 e3840741 e3840751 e3840761 e3840771 e3840781 e3840791;
array e384072 e3840712 e3840722 e3840732 e3840742 e3840752 e3840762 e3840772 e3840782 e3840792;
array e384073 e3840713 e3840723 e3840733 e3840743 e3840753 e3840763 e3840773 e3840783 e3840793;
array e384074 e3840714 e3840724 e3840734 e3840744 e3840754 e3840764 e3840774 e3840784 e3840794;
array e384075 e3840715 e3840725 e3840735 e3840745 e3840755 e3840765 e3840775 e3840785 e3840795;

array e102051 e1020512 e1020522 e1020532 e1020542 e1020552 e1020562 e1020572 e1020582 e1020592;
array e102052 e1020513 e1020523 e1020533 e1020543 e1020553 e1020563 e1020573 e1020583 e1020593;
array e102053 e1020514 e1020524 e1020534 e1020544 e1020554 e1020564 e1020574 e1020584 e1020594;
array e102054 e1020515 e1020525 e1020535 e1020545 e1020555 e1020565 e1020575 e1020585 e1020595;
array e102055 e1020516 e1020526 e1020536 e1020546 e1020556 e1020566 e1020576 e1020586 e1020596;

array e102141 e1021412 e1021422 e1021432 e1021442 e1021452 e1021462 e1021472 e1021482 e1021492;
array e102142 e1021413 e1021423 e1021433 e1021443 e1021453 e1021463 e1021473 e1021483 e1021493;
array e102143 e1021414 e1021424 e1021434 e1021444 e1021454 e1021464 e1021474 e1021484 e1021494;
array e102144 e1021415 e1021425 e1021435 e1021445 e1021455 e1021465 e1021475 e1021485 e1021495;
array e102145 e1021416 e1021426 e1021436 e1021446 e1021456 e1021466 e1021476 e1021486 e1021496;

array e10214aaa1 e10214aaa12 e10214aaa22 e10214aaa32 e10214aaa42 e10214aaa52 e10214aaa62 e10214aaa72
e10214aaa82 e10214aaa92;
array e10214aaa2 e10214aaa13 e10214aaa23 e10214aaa33 e10214aaa43 e10214aaa53 e10214aaa63 e10214aaa73
e10214aaa83 e10214aaa93;
array e10214aaa3 e10214aaa14 e10214aaa24 e10214aaa34 e10214aaa44 e10214aaa54 e10214aaa64 e10214aaa74
e10214aaa84 e10214aaa94;
array e10214aaa4 e10214aaa15 e10214aaa25 e10214aaa35 e10214aaa45 e10214aaa55 e10214aaa65 e10214aaa75
e10214aaa85 e10214aaa95;
array e10214aaa5 e10214aaa16 e10214aaa26 e10214aaa36 e10214aaa46 e10214aaa56 e10214aaa66 e10214aaa76
e10214aaa86 e10214aaa96;

array e10214aab1 e10214aab12 e10214aab22 e10214aab32 e10214aab42 e10214aab52 e10214aab62 e10214aab72
e10214aab82 e10214aab92;
array e10214aab2 e10214aab13 e10214aab23 e10214aab33 e10214aab43 e10214aab53 e10214aab63 e10214aab73
e10214aab83 e10214aab93;
array e10214aab3 e10214aab14 e10214aab24 e10214aab34 e10214aab44 e10214aab54 e10214aab64 e10214aab74
e10214aab84 e10214aab94;
array e10214aab4 e10214aab15 e10214aab25 e10214aab35 e10214aab45 e10214aab55 e10214aab65 e10214aab75
e10214aab85 e10214aab95;
array e10214aab5 e10214aab16 e10214aab26 e10214aab36 e10214aab46 e10214aab56 e10214aab66 e10214aab76
e10214aab86 e10214aab96;

array e10214b1 e10214b12 e10214b22 e10214b32 e10214b42 e10214b52 e10214b62 e10214b72 e10214b82
e10214b92;
array e10214b2 e10214b13 e10214b23 e10214b33 e10214b43 e10214b53 e10214b63 e10214b73 e10214b83
e10214b93;

```

Appendix 2: Employment Variable Creation

```
array e10214b3 e10214b14 e10214b24 e10214b34 e10214b44 e10214b54 e10214b64 e10214b74 e10214b84  
    e10214b94;  
array e10214b4 e10214b15 e10214b25 e10214b35 e10214b45 e10214b55 e10214b65 e10214b75 e10214b85  
    e10214b95;  
array e10214b5 e10214b16 e10214b26 e10214b36 e10214b46 e10214b56 e10214b66 e10214b76 e10214b86  
    e10214b96;  
  
array e226041 e2260411 e2260421 e2260431 e2260441 e2260451 e2260461 e2260471 e2260481 e2260491;  
array e226042 e2260412 e2260422 e2260432 e2260442 e2260452 e2260462 e2260472 e2260482 e2260492;  
array e226043 e2260413 e2260423 e2260433 e2260443 e2260453 e2260463 e2260473 e2260083 e2260493;  
array e226044 e2260414 e2260424 e2260434 e2260444 e2260454 e2260464 e2260474 e2260484 e2260494;  
array e226045 e2260415 e2260425 e2260435 e2260445 e2260455 e2260465 e2260475 e2260485 e2260495;  
array e226046 e2260416 e2260426 e2260436 e2260446 e2260456 e2260466 e2260476 e2260486 e2260496;  
array e226047 e2260417 e2260427 e2260437 e2260447 e2260457 e2260467 e2260477 e2260487 e2260497;  
array e226048 e2260418 e2260428 e2260438 e2260448 e2260458 e2260468 e2260478 e2260088 e2260498;  
array e226049 e2260419 e2260429 e2260439 e2260449 e2260459 e2260469 e2260479 e2260489 e2260499;  
array e226040 e2260410 e2260420 e2260430 e2260440 e2260450 e2260460 e2260470 e2260480 e2260490;
```

```
array e102251 e1022511 e1022521 e1022531 e1022541 e1022551 e1022561 e1022571 e1022581 e1022591;  
array e102252 e1022512 e1022522 e1022532 e1022542 e1022552 e1022562 e1022572 e1022582 e1022592;  
array e102253 e1022513 e1022523 e1022533 e1022543 e1022553 e1022563 e1022573 e2260083 e1022593;  
array e102254 e1022514 e1022524 e1022534 e1022544 e1022554 e1022564 e1022574 e1022584 e1022594;  
array e102255 e1022515 e1022525 e1022535 e1022545 e1022555 e1022565 e1022575 e1022585 e1022595;  
array e102256 e1022516 e1022526 e1022536 e1022546 e1022556 e1022566 e1022576 e1022586 e1022596;  
array e102257 e1022517 e1022527 e1022537 e1022547 e1022557 e1022567 e1022577 e1022587 e1022597;  
array e102258 e1022518 e1022528 e1022538 e1022548 e1022558 e1022568 e1022578 e2260088 e1022598;  
array e102259 e1022519 e1022529 e1022539 e1022549 e1022559 e1022569 e1022579 e1022589 e1022599;  
array e102250 e1022510 e1022520 e1022530 e1022540 e1022550 e1022560 e1022570 e1022580 e1022590;
```

```
***** We create the hourly wages. First, the start wages, then the end wages *****  
***** Start wages for the youth *****
```

```
/* For each time unit, the information could either be given in the earlier part (from e19200) or in the latet part (from  
e83100). And it also depends on if the respondent have other compensation or not. */  
/*Set up the hourly wage for youths who report an hourly wage*/  
hrwage01=-4;    hrwage02=-4;    hrwage03=-4;    hrwage04=-4;    hrwage05=-4;  
hrwage06=-4;    hrwage07=-4;    hrwage08=-4;    hrwage09=-4;
```

```
array hrwage hrwage01 hrwage02 hrwage03 hrwage04 hrwage05 hrwage06 hrwage07 hrwage08 hrwage09;
```

```
do i=1 to 9;
```

```
if e19200[i]=1 then do;  
  if (e22900[i]>=0) then hrwage[i]=e22900[i];  
  if (e23000[i]>=0) then hrwage[i]=e23000[i];  
  if (e22900[i]=-2 or e23000[i]=-2) then hrwage[i]=e23200[i];  
  if (e22900[i]=-3 or e23000[i]=-3) then hrwage[i]=e23200[i];  
  if (e22900[i]=-1 or e23000[i]=-1) then hrwage[i]=-1;  
  if e23900[i]>=0 then hrwage[i]=e23900[i];  
end;
```

```
if e83100[i]=1 then do;  
  if (e86800[i]>=0) then hrwage[i]=e86800[i];  
  if (e86900[i]>=0) then hrwage[i]=e86900[i];  
  if (e86800[i]=-2 or e86900[i]=-2) then hrwage[i]=e87100[i];  
  if (e86800[i]=-3 or e86900[i]=-3) then hrwage[i]=e87100[i];
```

Appendix 2: Employment Variable Creation

```
if (e86800[i]=-1 or e86900[i]=-1) then hrwage[i]=-1;
  if e87800[i]>=0 then hrwage[i]=e87800[i];
end;

end;

/*Set up the hourly wage for youths who report their wage in daily units.*/
daily01=-4;      daily02=-4;      daily03=-4;      daily04=-4;      daily05=-4;
daily06=-4;      daily07=-4;      daily08=-4;      daily09=-4;

array daily daily01 daily02 daily03 daily04 daily05 daily06 daily07 daily08 daily09;

do i=1 to 9;

/* daily start wage divided by the number of hours worked per week*/

if e19200[i]=2 then do;

/*no compensation*/
  if (e33400[i]>=0 and e34402[i]>0 and e34402ab[i]>0) then daily[i]=(e33400[i]*e34402ab[i]/e34402[i]);
  if (e33400[i]=-2 and e33600[i]>=0 and e34402[i]>0 and e34402ab[i]>0) then
    daily[i]=(e33600[i]*e34402ab[i]/e34402[i]);
  if e34400[i]>=0 and e34402[i]>0 and e34402ab[i]>0 then daily[i]=e34400[i]*e34402ab[i]/e34402[i];
/* missing value*/
  if -4<e34402[i]<0 then daily[i]=e34402[i];

/*received compensation*/
/*without overtime*/
if (e33500[i]>=0 and e34402[i]>0 and e34402ab[i]>0) then daily[i]=(e33500[i]*e34402ab[i]/e34402[i]);
if (e33500[i]=-2 and e33600[i]>=0 and e34402[i]>0 and e34402ab[i]>0) then
  daily[i]=(e33600[i]*e34402ab[i]/e34402[i]);
if e34400[i]>=0 and e34402[i]>0 and e34402ab[i]>0 then daily[i]=e34400[i]*e34402ab[i]/e34402[i];
/*missing value*/
if -4<e34402[i]<0 then daily[i]=e34402[i];
if (e33400[i]>=0 or e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34402[i]=0 then daily[i]=-3;

if -4<e34402ab[i]<0 then daily[i]=e34402ab[i];
if e34402ab[i]=0 then daily[i]=-3;

/*with overtime*/
if (e33500[i]>=0 and e34428[i]>0 and e34430[i]>0) then daily[i]=(e33500[i]*e34430[i]/e34428[i]);
if (e33500[i]=-2 and e33600[i]>=0 and e34428[i]>0 and e34430[i]>0) then daily[i]=(e33600[i]*e34430[i]/e34428[i]);
if e34400[i]>=0 and e34428[i]>0 and e34430[i]>0 then daily[i]=e34400[i]*e34430[i]/e34428[i];
/*missing value*/
if -4<e34428[i]<0 then daily[i]=e34428[i];
if (e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34428[i]=0 then daily[i]=-3;
if -4<e34430[i]<0 then daily[i]=e34430[i];
if e34430[i]=0 then daily[i]=-3;

/*if still paid hourly...*/
if e36200[i]>=0 then daily[i]=e36200[i];
if e36100[i]>=0 then daily[i]=e36100[i];

end;

if e83100[i]=2 then do;
```

```

/*no compensation*/
if (e97300[i]>=0 and e98402[i]>0 and e98402d[i]>0) then daily[i]=(e97300[i]*e98402d[i]/e98402[i]);
if (e97300[i]=-2 and e97500[i]>=0 and e98402[i]>0 and e98402d[i]>0) then
    daily[i]=(e97500[i]*e98402d[i]/e98402[i]);
if e98300[i]>=0 and e98402[i]>0 and e98402d[i]>0 then daily[i]=e98300[i]*e98402d[i]/e98402[i];
/* missing value*/
if -4<e98402[i]<0 then daily[i]=e98402[i];

/*received compensation*/
/*with overtime */
if (e97400[i]>=0 and e98429[i]>0 and e98429l[i]>0) then daily[i]=(e97400[i]*e98429l[i]/e98429[i]);
if (e97400[i]=-2 and e97500[i]>=0 and e98429[i]>0 and e98429l[i]>0) then daily[i]=(e97500[i]*e98429l[i]/e98429[i]);
if e98300[i]>=0 and e98429[i]>0 and e98429l[i]>0 then daily[i]=e98300[i]*e98429l[i]/e98429[i];
/*missing value*/
if -4<e98429[i]<0 then daily[i]=e98429[i];
if (e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98429[i]=0 then daily[i]=-3;
if -4<e98429l[i]<0 then daily[i]=e98429l[i];
if e98429l[i]=0 then daily[i]=-3;

/* without overtime */
if (e97400[i]>=0 and e98402[i]>0 and e98402d[i]>0) then daily[i]=(e97400[i]*e98402d[i]/e98402[i]);
if (e97400[i]=-2 and e97500[i]>=0 and e98402[i]>0 and e98402d[i]>0) then
    daily[i]=(e97500[i]*e98402d[i]/e98402[i]);
if e98300[i]>=0 and e98402[i]>0 and e98402d[i]>0 then daily[i]=e98300[i]*e98402d[i]/e98402[i];
/*missing value*/
if -4<e98402[i]<0 then daily[i]=e98402[i];
if (e97300[i]>=0 or e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98402[i]=0 then daily[i]=-3;
if -4<e98402d[i]<0 then daily[i]=e98402d[i];
if e98402d[i]=0 then daily[i]=-3;

/*if still paid hourly...*/
if e100100[i]>=0 then daily[i]=e100100[i];
if e100000[i]>=0 then daily[i]=e100000[i];

end;
end;

/*set up the hourly wage for youths who report their wage in weekly units.*/
weekly01=-4;      weekly02=-4;      weekly03=-4;      weekly04=-4;      weekly05=-4;
weekly06=-4;      weekly07=-4;      weekly08=-4;      weekly09=-4;

array weekly weekly01 weekly02 weekly03 weekly04 weekly05 weekly06 weekly07 weekly08 weekly09;

do i=1 to 9;

/* weekly start wage divided by the number of hours worked per week*/

if e19200[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999,-2) then do;

/*no compensation*/
if (e33400[i]>=0 and e34402[i]>0) then weekly[i]=(e33400[i]/e34402[i]);
if (e33400[i]=-2 and e33600[i]>=0 and e34402[i]>0) then weekly[i]=(e33600[i]/e34402[i]);
if e34400[i]>=0 and e34402[i]>0 then weekly[i]=e34400[i]/e34402[i];
/*missing value*/
if -4<e34402[i]<0 then weekly[i]=e34402[i];

/*received compensation*/

```

```

/*with overtime*/
if (e33500[i]>=0 and e34428[i]>0) then weekly[i]=(e33500[i]/e34428[i]);
if (e33500[i]=-2 and e33600[i]>=0 and e34428[i]>0) then weekly[i]=(e33600[i]/e34428[i]);
if e34400[i]>=0 and e34428[i]>0 then weekly[i]=e34400[i]/e34428[i];
/*missing value*/
if -4<e34428[i]<0 then weekly[i]=e34428[i];
if (e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34428[i]=0 then weekly[i]=-3;

/*without overtime*/
if (e33500[i]>=0 and e34402[i]>0) then weekly[i]=(e33500[i]/e34402[i]);
if (e33500[i]=-2 and e33600[i]>=0 and e34402[i]>0) then weekly[i]=(e33600[i]/e34402[i]);
if e34400[i]>=0 and e34402[i]>0 then weekly[i]=e34400[i]/e34402[i];
/*missing value*/
if -4<e34402[i]<0 then weekly[i]=e34402[i];
if (e33400[i]>=0 or e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34402[i]=0 then weekly[i]=-3;

/*if still paid hourly...*/
if e36200[i]>=0 then weekly[i]=e36200[i];
if e36100[i]>=0 then weekly[i]=e36100[i];

end;

if e83100[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999,-2) then do;

/*no compensation*/
if (e97300[i]>=0 and e98402[i]>0) then weekly[i]=(e97300[i]/e98402[i]);
if (e97300[i]=-2 and e97500[i]>=0 and e98402[i]>0) then weekly[i]=(e97500[i]/e98402[i]);
if e98300[i]>=0 and e98402[i]>0 then weekly[i]=e98300[i]/e98402[i];
/*missing value*/
if -4<e98402[i]<0 then weekly[i]=e98402[i];

/*received compensation*/
/*with overtime*/
if (e97400[i]>=0 and e98429[i]>0) then weekly[i]=(e97400[i]/e98429[i]);
if (e97400[i]=-2 and e97500[i]>=0 and e98429[i]>0) then weekly[i]=(e97500[i]/e98429[i]);
if e98300[i]>=0 and e98429[i]>0 then weekly[i]=e98300[i]/e98429[i];
/*missing value*/
if -4<e98429[i]<0 then weekly[i]=e98429[i];
if (e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98429[i]=0 then weekly[i]=-3;

/*without overtime*/
if (e97400[i]>=0 and e98402[i]>0) then weekly[i]=(e97400[i]/e98402[i]);
if (e97400[i]=-2 and e97500[i]>=0 and e98402[i]>0) then weekly[i]=(e97500[i]/e98402[i]);
if e98300[i]>=0 and e98402[i]>0 then weekly[i]=e98300[i]/e98402[i];
/*missing value*/
if -4<e98402[i]<0 then weekly[i]=e98402[i];
if (e97300[i]>=0 or e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98402[i]=0 then weekly[i]=-3;

/*if still paid hourly...*/
if e100100[i]>=0 then weekly[i]=e100100[i];
if e100000[i]>=0 then weekly[i]=e100000[i];

end;
end;

/*set up the hourly wage for youths who report their wage in biweekly units.*/

```

Appendix 2: Employment Variable Creation

```
biwkly01=-4;      biwkly02=-4;      biwkly03=-4;      biwkly04=-4;      biwkly05=-4;
biwkly06=-4;      biwkly07=-4;      biwkly08=-4;      biwkly09=-4;

array biwkly biwkly01 biwkly02 biwkly03 biwkly04 biwkly05 biwkly06 biwkly07 biwkly08 biwkly09;

do i=1 to 9;

/* biwkly start wage divided by the number of hours worked per week*/

if e19200[i]=4 then do;

/*no compensation*/
if (e33400[i]>=0 and e34402[i]>0) then biwkly[i]=e33400[i]/(2*e34402[i]);
if (e33400[i]=-2 and e33600[i]>=0 and e34402[i]>0) then biwkly[i]=e33600[i]/(2*e34402[i]);
if e34400[i]>=0 and e34402[i]>0 then biwkly[i]=e34400[i]/(2*e34402[i]);
/*missing value*/
if -4<e34402[i]<0 then biwkly[i]=e34402[i];

/*received compensation*/
/*with overtime*/
if (e33500[i]>=0 and e34428[i]>0) then biwkly[i]=e33500[i]/(2*e34428[i]);
if (e33500[i]=-2 and e33600[i]>=0 and e34428[i]>0) then biwkly[i]=e33600[i]/(2*e34428[i]);
if e34400[i]>=0 and e34428[i]>0 then biwkly[i]=e34400[i]/(2*e34428[i]);
/*missing value*/
if -4<e34428[i]<0 then biwkly[i]=e34428[i];
if (e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34428[i]=0 then biwkly[i]=-3;

/*without overtime*/
if (e33500[i]>=0 and e34402[i]>0) then biwkly[i]=e33500[i]/(2*e34402[i]);
if (e33500[i]=-2 and e33600[i]>=0 and e34402[i]>0) then biwkly[i]=e33600[i]/(2*e34402[i]);
if e34400[i]>=0 and e34402[i]>0 then biwkly[i]=e34400[i]/(2*e34402[i]);
/*missing value*/
if -4<e34402[i]<0 then biwkly[i]=e34402[i];
if (e33400[i]>=0 or e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34402[i]=0 then biwkly[i]=-3;

/*if still paid hourly...*/
if e36200[i]>=0 then biwkly[i]=e36200[i];
if e36100[i]>=0 then biwkly[i]=e36100[i];

end;

if e83100[i]=4 then do;

/*no compensation*/
if (e97300[i]>=0 and e98402[i]>0) then biwkly[i]=e97300[i]/(2*e98402[i]);
if (e97300[i]=-2 and e97500[i]>=0 and e98402[i]>0) then biwkly[i]=e97500[i]/(2*e98402[i]);
if e98300[i]>=0 and e98402[i]>0 then biwkly[i]=e98300[i]/(2*e98402[i]);
/*missing value*/
if -4<e98402[i]<0 then biwkly[i]=e98402[i];

/*received compensation*/
/*with overtime*/
if (e97400[i]>=0 and e98429[i]>0) then biwkly[i]=e97400[i]/(2*e98429[i]);
if (e97400[i]=-2 and e97500[i]>=0 and e98429[i]>0) then biwkly[i]=e97500[i]/(2*e98429[i]);
if e98300[i]>=0 and e98429[i]>0 then biwkly[i]=e98300[i]/(2*e98429[i]);
/*missing value*/
if -4<e98429[i]<0 then biwkly[i]=e98429[i];
```

Appendix 2: Employment Variable Creation

```
if (e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98429[i]=0 then biwkly[i]=-3;

/*without overtime*/
if (e97400[i]>=0 and e98402[i]>0) then biwkly[i]=e97400[i]/(2*e98402[i]);
if (e97400[i]=-2 and e97500[i]>=0 and e98402[i]>0) then biwkly[i]=e97500[i]/(2*e98402[i]);
if e98300[i]>=0 and e98402[i]>0 then biwkly[i]=e98300[i]/(2*e98402[i]);
/*missing value*/
if -4<e98402[i]<0 then biwkly[i]=e98402[i];
if (e97300[i]>=0 or e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98402[i]=0 then biwkly[i]=-3;

/*if still paid hourly...*/
if e100100[i]>=0 then biwkly[i]=e100100[i];
if e100000[i]>=0 then biwkly[i]=e100000[i];

end;
end;

/*set up the hourly wage for youths who report their wage in monthly units.*/
month01=-4; month02=-4; month03=-4; month04=-4; month05=-4;
month06=-4; month07=-4; month08=-4; month09=-4;

array month month01 month02 month03 month04 month05 month06 month07 month08 month09;

do i=1 to 9;

/* month start wage divided by the number of hours worked per week*/

if e19200[i]=5 then do;

/*no compensation*/
if (e33400[i]>=0 and e34402[i]>0) then month[i]=e33400[i]/(4.3*e34402[i]);
if (e33400[i]=-2 and e33600[i]>=0 and e34402[i]>0) then month[i]=e33600[i]/(4.3*e34402[i]);
if e34400[i]>=0 and e34402[i]>0 then month[i]=e34400[i]/(4.3*e34402[i]);
/*missing value*/
if -4<e34402[i]<0 then month[i]=e34402[i];

/*received compensation*/
/*with overtime*/
if (e33500[i]>=0 and e34428[i]>0) then month[i]=e33500[i]/(4.3*e34428[i]);
if (e33500[i]=-2 and e33600[i]>=0 and e34428[i]>0) then month[i]=e33600[i]/(4.3*e34428[i]);
if e34400[i]>=0 and e34428[i]>0 then month[i]=e34400[i]/(4.3*e34428[i]);
/*missing value*/
if -4<e34428[i]<0 then month[i]=e34428[i];
if (e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34428[i]=0 then month[i]=-3;

/*without overtime*/
if (e33500[i]>=0 and e34402[i]>0) then month[i]=e33500[i]/(4.3*e34402[i]);
if (e33500[i]=-2 and e33600[i]>=0 and e34402[i]>0) then month[i]=e33600[i]/(4.3*e34402[i]);
if e34400[i]>=0 and e34402[i]>0 then month[i]=e34400[i]/(4.3*e34402[i]);
/*missing value*/
if -4<e34402[i]<0 then month[i]=e34402[i];
if (e33400[i]>=0 or e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34402[i]=0 then month[i]=-3;

/*if still paid hourly...*/
if e36200[i]>=0 then month[i]=e36200[i];
if e36100[i]>=0 then month[i]=e36100[i];
```

Appendix 2: Employment Variable Creation

```
end;

if e83100[i]=5 then do;

/*no compensation*/
if (e97300[i]>=0 and e98402[i]>0) then month[i]=e97300[i]/(4.3*e98402[i]);
if (e97300[i]=-2 and e97500[i]>=0 and e98402[i]>0) then month[i]=e97500[i]/(4.3*e98402[i]);
if e98300[i]>=0 and e98402[i]>0 then month[i]=e98300[i]/(4.3*e98402[i]);
/*missing value*/
if -4<e98402[i]<0 then month[i]=e98402[i];

/*received compensation*/
/*with overtime*/
if (e97400[i]>=0 and e98429[i]>0) then month[i]=e97400[i]/(4.3*e98429[i]);
if (e97400[i]=-2 and e97500[i]>=0 and e98429[i]>0) then month[i]=e97500[i]/(4.3*e98429[i]);
if e98300[i]>=0 and e98429[i]>0 then month[i]=e98300[i]/(4.3*e98429[i]);
/*missing value*/
if -4<e98429[i]<0 then month[i]=e98429[i];
if (e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98429[i]=0 then month[i]=-3;

/*without overtime*/
if (e97400[i]>=0 and e98402[i]>0) then month[i]=e97400[i]/(4.3*e98402[i]);
if (e97400[i]=-2 and e97500[i]>=0 and e98402[i]>0) then month[i]=e97500[i]/(4.3*e98402[i]);
if e98300[i]>=0 and e98402[i]>0 then month[i]=e98300[i]/(4.3*e98402[i]);
/*missing value*/
if -4<e98402[i]<0 then month[i]=e98402[i];
if (e97300[i]>=0 or e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98402[i]=0 then month[i]=-3;

/*if still paid hourly...*/
if e100100[i]>=0 then month[i]=e100100[i];
if e100000[i]>=0 then month[i]=e100000[i];

end;
end;

/*set up the hourly wage for youths who report their wage in annual units.*/
annual01=-4;      annual02=-4;      annual03=-4;      annual04=-4;      annual05=-4;
annual06=-4;      annual07=-4;      annual08=-4;      annual09=-4;

array annual annual01 annual02 annual03 annual04 annual05 annual06 annual07 annual08 annual09;

do i=1 to 9;

/* annual start wage divided by the number of hours worked per week*/

if e19200[i]=6 then do;

/*no compensation*/
if (e33400[i]>=0 and e34402[i]>0 and e35600[i]>0) then annual[i]=e33400[i]/(e35600[i]*e34402[i]);
if (e33400[i]=-2 and e33600[i]>=0 and e34402[i]>0 and e35600[i]>0) then
    annual[i]=e33600[i]/(e35600[i]*e34402[i]);
if e34400[i]>=0 and e34402[i]>0 and e35600[i]>0 then annual[i]=e34400[i]/(e35600[i]*e34402[i]);
/*missing value*/
if -4<e34402[i]<0 then annual[i]=e34402[i];

/*received compensation*/
/*with overtime*/
```

Appendix 2: Employment Variable Creation

```

if (e33500[i]>=0 and e34428[i]>0 and e35600[i]>0) then annual[i]=e33500[i]/(e35600[i]*e34428[i]);
if (e33500[i]=-2 and e33600[i]>=0 and e34428[i]>0 and e35600[i]>0) then
    annual[i]=e33600[i]/(e35600[i]*e34428[i]);
if e34400[i]>=0 and e34428[i]>0 and e35600[i]>0 then annual[i]=e34400[i]/(e35600[i]*e34428[i]);
/*missing value*/
if -4<e34428[i]<0 then annual[i]=e34428[i];
if (e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34428[i]=0 then annual[i]=-3;

/*without overtime*/
if (e33500[i]>=0 and e34402[i]>0 and e35600[i]>0) then annual[i]=e33500[i]/(e35600[i]*e34402[i]);
if (e33500[i]=-2 and e33600[i]>=0 and e34402[i]>0 and e35600[i]>0) then
    annual[i]=e33600[i]/(e35600[i]*e34402[i]);
if e34400[i]>=0 and e34402[i]>0 and e35600[i]>0 then annual[i]=e34400[i]/(e35600[i]*e34402[i]);
/*missing value*/
if -4<e34402[i]<0 then annual[i]=e34402[i];
if (e33400[i]>=0 or e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34402[i]=0 then annual[i]=-3;
if -4<e35600[i]<0 then annual[i]=e35600[i];
if e35600[i]=0 then annual[i]=-3;

/*if still paid hourly...*/
if e36200[i]>=0 then annual[i]=e36200[i];
if e36100[i]>=0 then annual[i]=e36100[i];

end;

if e83100[i]=6 then do;

/*no compensation*/
if (e97300[i]>=0 and e98402[i]>0 and e99500[i]>0) then annual[i]=e97300[i]/(e99500[i]*e98402[i]);
if (e97300[i]=-2 and e97500[i]>=0 and e98402[i]>0 and e99500[i]>0) then
    annual[i]=e97500[i]/(e99500[i]*e98402[i]);
if e98300[i]>=0 and e98402[i]>0 and e99500[i]>0 then annual[i]=e98300[i]/(e99500[i]*e98402[i]);
/*missing value*/
if -4<e98402[i]<0 then annual[i]=e98402[i];

/*received compensation*/
/*with overtime*/
if (e97400[i]>=0 and e98429[i]>0 and e99500[i]>0) then annual[i]=e97400[i]/(e99500[i]*e98429[i]);
if (e97400[i]=-2 and e97500[i]>=0 and e98429[i]>0 and e99500[i]>0) then
    annual[i]=e97500[i]/(e99500[i]*e98429[i]);
if e98300[i]>=0 and e98429[i]>0 and e99500[i]>0 then annual[i]=e98300[i]/(e99500[i]*e98429[i]);
/*missing value*/
if -4<e99500[i]<0 then annual[i]=e99500[i];
if -4<e98429[i]<0 then annual[i]=e98429[i];
if (e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98429[i]=0 then annual[i]=-3;

/* without overtime*/
if (e97400[i]>=0 and e98402[i]>0 and e99500[i]>0) then annual[i]=e97400[i]/(e99500[i]*e98402[i]);
if (e97400[i]=-2 and e97500[i]>=0 and e98402[i]>0 and e99500[i]>0) then
    annual[i]=e97500[i]/(e99500[i]*e98402[i]);
if e98300[i]>=0 and e98402[i]>0 and e99500[i]>0 then annual[i]=e98300[i]/(e99500[i]*e98402[i]);
/*missing value*/
if -4<e99500[i]<0 then annual[i]=e99500[i];
if -4<e98402[i]<0 then annual[i]=e98402[i];
if (e97300[i]>=0 or e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98402[i]=0 then annual[i]=-3;
if -4<e99500[i]<0 then annual[i]=e99500[i];
if e99500[i]=0 then annual[i]=-3;

```

```

/*if still paid hourly...*/
if e100100[i]>=0 then annual[i]=e100100[i];
if e100000[i]>=0 then annual[i]=e100000[i];

end;
end;

/*set up the hourly wage for youths who report their wage in semim units.*/
semim01=-4;      semim02=-4;      semim03=-4;      semim04=-4;      semim05=-4;
semim06=-4;      semim07=-4;      semim08=-4;      semim09=-4;

array semim semim01 semim02 semim03 semim04 semim05 semim06 semim07 semim08 semim09;

do i=1 to 9;

/* semim start wage divided by the number of hours worked per week*/

if e19200[i]=8 then do;

/*no compensation*/
if (e33400[i]>=0 and e34402[i]>0) then semim[i]=e33400[i]/(2.15*e34402[i]);
if (e33400[i]=-2 and e33600[i]>=0 and e34402[i]>0) then semim[i]=e33600[i]/(2.15*e34402[i]);
if e34400[i]>=0 and e34402[i]>0 then semim[i]=e34400[i]/(2.15*e34402[i]);
/*missing value*/
if -4<e34402[i]<0 then semim[i]=e34402[i];

/*received compensation*/
/*with overtime*/
if (e33500[i]>=0 and e34428[i]>0) then semim[i]=e33500[i]/(2.15*e34428[i]);
if (e33500[i]=-2 and e33600[i]>=0 and e34428[i]>0) then semim[i]=e33600[i]/(2.15*e34428[i]);
if e34400[i]>=0 and e34428[i]>0 then semim[i]=e34400[i]/(2.15*e34428[i]);
/*missing value*/
if -4<e34428[i]<0 then semim[i]=e34428[i];
if (e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34428[i]=0 then semim[i]=-3;

/*without overtime*/
if (e33500[i]>=0 and e34402[i]>0) then semim[i]=e33500[i]/(2.15*e34402[i]);
if (e33500[i]=-2 and e33600[i]>=0 and e34402[i]>0) then semim[i]=e33600[i]/(2.15*e34402[i]);
if e34400[i]>=0 and e34402[i]>0 then semim[i]=e34400[i]/(2.15*e34402[i]);
/*missing value*/
if -4<e34402[i]<0 then semim[i]=e34402[i];
if (e33400[i]>=0 or e33500[i]>=0 or e33600[i]>=0 or e34400[i]>=0) and e34402[i]=0 then semim[i]=-3;

/*if still paid hourly...*/
if e36200[i]>=0 then semim[i]=e36200[i];
if e36100[i]>=0 then semim[i]=e36100[i];

end;

if e83100[i]=8 then do;

/*no compensation*/
if (e97300[i]>=0 and e98402[i]>0) then semim[i]=e97300[i]/(2.15*e98402[i]);
if (e97300[i]=-2 and e97500[i]>=0 and e98402[i]>0) then semim[i]=e97500[i]/(2.15*e98402[i]);
if e98300[i]>=0 and e98402[i]>0 then semim[i]=e98300[i]/(2.15*e98402[i]);
/*missing value*/

```

```

if -4<e98402[i]<0 then semim[i]=e98402[i];

/*received compensation*/
/*with overtime*/
if (e97400[i]>=0 and e98429[i]>0) then semim[i]=e97400[i]/(2.15*e98429[i]);
if (e97400[i]=-2 and e97500[i]>=0 and e98429[i]>0) then semim[i]=e97500[i]/(2.15*e98429[i]);
if e98300[i]>=0 and e98429[i]>0 then semim[i]=e98300[i]/(2.15*e98429[i]);
/*missing value*/
if -4<e98429[i]<0 then semim[i]=e98429[i];
if (e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98429[i]=0 then semim[i]=-3;

/*without overtime*/
if (e97400[i]>=0 and e98402[i]>0) then semim[i]=e97400[i]/(2.15*e98402[i]);
if (e97400[i]=-2 and e97500[i]>=0 and e98402[i]>0) then semim[i]=e97500[i]/(2.15*e98402[i]);
if e98300[i]>=0 and e98402[i]>0 then semim[i]=e98300[i]/(2.15*e98402[i]);
/*missing value*/
if -4<e98402[i]<0 then semim[i]=e98402[i];
if (e97300[i]>=0 or e97400[i]>=0 or e97500[i]>=0 or e98300[i]>=0) and e98402[i]=0 then semim[i]=-3;

/*if still paid hourly...*/
if e100100[i]>=0 then semim[i]=e100100[i];
if e100000[i]>=0 then semim[i]=e100000[i];

end;
end;

/*set up the hourly wage for youths who report their wage by item or job.*/
item01=-4;      item02=-4;      item03=-4;      item04=-4;      item05=-4;
item06=-4;      item07=-4;      item08=-4;      item09=-4;

array item item01 item02 item03 item04 item05 item06 item07 item08 item09;

flag34428=0; flag98429=0;

do i=1 to 9;

if e19200[i] in (12,13) then do;

/* first, get the total pay for each item or job, we name this variable as pay. */
if e33400[i]>=0 then pay[i]=e33400[i]; /*no comp.*/
if e33500[i]>=0 then pay[i]=e33500[i]; /*with comp.*/
if e33600[i]>=0 then pay[i]=e33600[i]; /*if -2 for one of e33400 or e33500.*/
if e34400[i]>=0 then pay[i]=e34400[i]; /*if previous entry is incorrect.*/

/* without overtime, where e34402 will be the weekly hour info.*/
/* condition based on the time unit to finish per item.*/
if e34402aab[i]=1 and e34402aad[i]>0 then item[i]=(pay[i]/e34402aad[i])*60;
if e34402aab[i]=2 and e34402aad[i]>0 then item[i]=pay[i]/e34402aad[i];
if e34402aab[i]=3 and e34402aad[i]>0 and e34402ab[i]>=0 and e34402[i]>0 then
    item[i]=(pay[i]/e34402aad[i])*e34402ab[i]/e34402[i];
if e34402aab[i]=4 and e34402aad[i]>0 and e34402[i]>0 then item[i]=(pay[i]/e34402aad[i])/e34402[i];
if e34402aab[i]=5 and e34402aad[i]>0 and e34402[i]>0 then item[i]=(pay[i]/e34402aad[i])/(e34402[i]*4.3);
/* missing value */
if -4<e34402aab[i]<0 or -4<e34402aad[i]<=0 then item[i]=-3;
if e34402aab[i] in (3,4,5) and -4<e34402[i]<=0 then item[i]=-3;
if e34402aab[i]=3 and e34402ab[i]<0 then item[i]=-3;

```

Appendix 2: Employment Variable Creation

```
/* with overtime, where e34428 will be the weekly hour info. and there is no info. available to calculate hourly pay. */

if e212001[i]=1 then do;
  item[i]=-3;
  flag34428=1;
end;

/*if still paid hourly...*/
if e36200[i]>=0 then item[i]=e36200[i];
if e36100[i]>=0 then item[i]=e36100[i];

end;

if e83100[i] in (12,13) then do;

  /* first, get the total pay for each item or job, we name this variable as pay. */
  if e97300[i]>=0 then pay[i]=e97300[i]; /*no comp.*/
  if e97400[i]>=0 then pay[i]=e97400[i]; /*with comp.*/
  if e97500[i]>=0 then pay[i]=e97500[i]; /*if -2 for one of e33400 or e33500.*/
  if e98300[i]>=0 then pay[i]=e98300[i]; /*if previous entry is incorrect.*/

  /* without overtime, where e98402 will be the weekly hour info.*/
  /* condition based on the time unit to finish per item.*/
  if e98402caa[i]=1 and e98402cab[i]>0 then item[i]=(pay[i]/e98402cab[i])*60;
  if e98402caa[i]=2 and e98402cab[i]>0 then item[i]=pay[i]/e98402cab[i];
  if e98402caa[i]=3 and e98402cab[i]>0 and e98402d[i]>=0 and e98402[i]>0 then
    item[i]=(pay[i]/e98402cab[i])*e98402d[i]/e98402[i];
  if e98402caa[i]=4 and e98402cab[i]>0 and e98402[i]>0 then item[i]=(pay[i]/e98402cab[i])/e98402[i];
  if e98402caa[i]=5 and e98402cab[i]>0 and e98402[i]>0 then item[i]=(pay[i]/e98402cab[i])/(e98402[i]*4.3);
  /* missing value */
  if -4<e98402caa[i]<0 or -4<e98402cab[i]<=0 then item[i]=-3;
  if e98402caa[i] in (3,4,5) and -4<e98402[i]<=0 then item[i]=-3;
  if e98402caa[i]=3 and e98402d[i]<0 then item[i]=-3;

  /* with overtime, where e98429 will be the weekly hour info. and there is no info. available to calculate hourly pay.*/

  if e83100[i]=1 then do; item[i]=-3; flag98429=1; end;

  /*if still paid hourly...*/
  if e100100[i]>=0 then item[i]=e100100[i];
  if e100000[i]>=0 then item[i]=e100000[i];

end;
end;

/*for the people who report other time units.*/
array otherf otherf1-otherf9;

do i=1 to 9; otherf[i]=0; end;

do i=1 to 9;

if e19200[i] in (7,9,12,13,15,16,17,14,18,19,21,22,23,24,25,26,28) or e83100[i] in
(7,9,12,13,15,16,17,14,18,19,21,22,23,24,25,26,28) then otherf[i]=otherf[i]+1;
end;
```

```

***** create the hourly rate of pay based on the start wage *****/
hrwg01=-4;      hrwg02=-4;      hrwg03=-4;      hrwg04=-4;      hrwg05=-4;
hrwg06=-4;      hrwg07=-4;      hrwg08=-4;      hrwg09=-4;

array hrwg hrwg01 hrwg02 hrwg03 hrwg04 hrwg05 hrwg06 hrwg07 hrwg08 hrwg09;

/* report hourly wage to be 0 for the family business and to be -3 for the poorly reported time units.*/
do i=1 to 9;
  if e19200[i] in (9,14) or e83100[i] in (9,14) then hrwg[i]=0;
end;

/* report hourly rate -1 or -2 if amount is -1 or -2*/
do i=1 to 9;
  if e19200[i] in (2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) then do;
    if e33400[i]=-1 or e33500[i]=-1 then hrwg[i]=-1;
    if (e33400[i]=-2 and e33600[i]=-2) or (e33500[i]=-2 and e33600[i]=-2) then hrwg[i]=-2;
    if e33400[i]=-3 or e33500[i]=-3 then hrwg[i]=-3;
  end;

  if e83100[i] in (2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) then do;
    if e97300[i]=-1 or e97400[i]=-1 then hrwg[i]=-1;
    if (e97300[i]=-2 and e97500[i]=-2) or (e97400[i]=-2 and e97500[i]=-2) then hrwg[i]=-2;
    if e97300[i]=-3 or e97400[i]=-3 then hrwg[i]=-3;
  end;
end;

/* report hourly rate -3 if no hours reported*/
do i=1 to 9;
  if e212001[i]=1 and e34428[i]=-4 then hrwg[i]=-3;
end;

do i=1 to 9;
  if annual[i] ge 0 then hrwg[i]=annual[i];      if month[i] ge 0 then hrwg[i]=month[i];
  if biwkly[i] ge 0 then hrwg[i]=biwkly[i];      if weekly[i] ge 0 then hrwg[i]=weekly[i];
  if daily[i] ge 0 then hrwg[i]=daily[i];      if hrwage[i] ge 0 then hrwg[i]=hrwage[i];
  if semim[i] ge 0 then hrwg[i]=semim[i];      if item[i] ge 0 then hrwg[i]=item[i];
  if hrwage[i] eq -1 or daily[i]=-1 or weekly[i] eq -1 or biwkly[i] eq -1 or month[i] eq -1 or annual[i] eq -1 or semim[i]=-1
    or item[i]=-1 then hrwg[i]=-1;
  if hrwage[i] eq -2 or daily[i]=-2 or weekly[i] eq -2 or biwkly[i] eq -2 or month[i] eq -2 or annual[i] eq -2 or semim[i]=-2
    or item[i]=-2 then hrwg[i]=-2;
  if hrwage[i] eq -3 or daily[i]=-3 or weekly[i] eq -3 or biwkly[i] eq -3 or month[i] eq -3 or annual[i] eq -3 or semim[i]=-3
    or item[i]=-3 then hrwg[i]=-3;
end;

/* report the corrected wage if the correction is made */

do i=1 to 9;
  if e226041[i]=1 and e226042[i]=0 then do; /* rate incorrect but hours correct*/

    /* no overtime, use e34402 for hours */
    if e22609[i]=1 and e22626[i]>=0 then hrwg[i]=e22626[i];
    if e22609[i]=2 and e34402[i]>0 and e22626e[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]*e22626e[i]/e34402[i];
    if e22609[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999,-1,-2) and e34402[i]>0 and e22626[i]>=0 then
      hrwg[i]=e22626[i]/e34402[i];
    if e22609[i]=4 and e34402[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/(2*e34402[i]);
  end;
end;

```

Appendix 2: Employment Variable Creation

```

if e22609[i]=5 and e34402[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/(4.3*e34402[i]);
if e22609[i]=6 and e34402[i]>0 and e35600[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/(e35600[i]*e34402[i]);
if e22609[i]=8 and e34402[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/(2.15*e34402[i]);
if e22609[i] in (9,14) then hrwg[i]=0;
if e22609[i]=2 and e22626e[i] le 0 then hrwg[i]=-3;
if e22609[i]=2 and -4<e22626e[i]<0 then hrwg[i]=e22626e[i];
if e22609[i]=6 and e35600[i] le 0 then hrwg[i]=-3;
if e22609[i]=6 and -4<e35600[i]<0 then hrwg[i]=e35600[i];
if e22609[i] in (12,13) and e22626[i]>=0 then do;
  /* condition based on the time unit to finish per hrwg. */
  if e22626c[i]=1 and e22626cb[i]>0 then hrwg[i]=(e22626[i]/e22626cb[i])*60;
  if e22626c[i]=2 and e22626cb[i]>0 then hrwg[i]=e22626[i]/e22626cb[i];
  if e22626c[i]=3 and e22626cb[i]>0 and e22626e[i]>=0 and e34402[i]>0 then
    hrwg[i]=(e22626[i]/e22626cb[i])*e22626e[i]/e34402[i];
  if e22626c[i]=4 and e22626cb[i]>0 and e34402[i]>0 then hrwg[i]=(e22626[i]/e22626cb[i])/e34402[i];
  if e22626c[i]=5 and e22626cb[i]>0 and e34402[i]>0 then hrwg[i]=(e22626[i]/e22626cb[i])/(e34402[i]*4.3);
  /* missing value */
  if -4<e22626c[i]<0 or -4<e22626cb[i]<=0 then hrwg[i]=-3;
  if e22626c[i] in (3,4,5) and -4<e34402[i]<=0 then hrwg[i]=-3;
  if e22626c[i]=3 and e22626e[i]<0 then hrwg[i]=-3;
end;

if e22609[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e22626[i]<0 then
  hrwg[i]=e22626[i];
if e22609[i] in (2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then hrwg[i]=-
  3;
if e22609[i] in (2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
  hrwg[i]=e34402[i];

/* overtime, use e34428 for hours*/
if e22609[i]=1 and e22626[i]>=0 then hrwg[i]=e22626[i];
if e22609[i]=2 and e34428[i]>0 and e22626e[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]*e22626e[i]/e34428[i];
if e22609[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, -1,-2,999) and e34428[i]>0 and e22626[i]>=0 then
  hrwg[i]=e22626[i]/e34428[i];
if e22609[i]=4 and e34428[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/(2*e34428[i]);
if e22609[i]=5 and e34428[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/(4.3*e34428[i]);
if e22609[i]=6 and e34428[i]>0 and e35600[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/(e35600[i]*e34428[i]);
if e22609[i]=8 and e34428[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/(2.15*e34428[i]);
if e22609[i] in (12,13) and e22626[i]>=0 then do;
  /* condition based on the time unit to finish per hrwg. */
  if e22626c[i]=3 and e22626cb[i]>0 and e22626e[i]>=0 and e34428[i]>0 then
    hrwg[i]=(e22626[i]/e22626cb[i])*e22626e[i]/e34428[i];
  if e22626c[i]=4 and e22626cb[i]>0 and e34428[i]>0 then hrwg[i]=(e22626[i]/e22626cb[i])/e34428[i];
  if e22626c[i]=5 and e22626cb[i]>0 and e34428[i]>0 then hrwg[i]=(e22626[i]/e22626cb[i])/(e34428[i]*4.3);
  /* missing value */
  if e22626c[i] in (3,4,5) and -4<e34428[i]<=0 then hrwg[i]=-3;
end;

if e22609[i] in (2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34428[i]=0 then hrwg[i]=-
  3;
if e22609[i] in (2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34428[i]<0 then
  hrwg[i]=e34428[i];
end;

if e226041[i]=0 and e226042[i]=1 then do; /*rate correct but hours incorrect*/
  if e19200[i] ne 1 and e22610[i]>0 and e34402[i]>0 and hrwg[i]>=0 then hrwg[i]=hrwg[i]*e34402[i]/e22610[i];
  if e19200[i] ne 1 and e22610[i]>0 and e34428[i]>0 and hrwg[i]>=0 then hrwg[i]=hrwg[i]*e34428[i]/e22610[i];

```

```

if e19200[i] ne 1 and e22610[i]=0 then hrwg[i]=-3;
if e19200[i] ne 1 and -4<e22610[i]<0 then hrwg[i]=e22610[i];
end;

if e226041[i]=1 and e226042[i]=1 then do; /* neither rate nor hours is correct*/
  if e22609[i]=1 and e22626[i]>=0 then hrwg[i]=e22626[i];
  if e22609[i]=2 and e22610[i]>0 and e22626e[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]*e22626e[i]/e22610[i];
  if e22609[i]=3 and e22610[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/e22610[i];
  if e22609[i]=4 and e22610[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/(2*e22610[i]);
  if e22609[i]=5 and e22610[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/(4.3*e22610[i]);
  if e22609[i]=6 and e22610[i]>0 and e35600[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/(e35600[i]*e22610[i]);
  if e22609[i]=8 and e22610[i]>0 and e22626[i]>=0 then hrwg[i]=e22626[i]/(2.15*e22610[i]);
  if e22609[i] in (9,14) then hrwg[i]=0;
  if e22609[i]=2 and e22626e[i] le 0 then hrwg[i]=-3;
  if e22609[i]=6 and e35600[i] le 0 then hrwg[i]=-3;
  if e22609[i] in (12,13) and e22626[i]>=0 then do;
    /* condition based on the time unit to finish per hrwg. */
    if e22626c[i]=1 and e22626cb[i]>0 then hrwg[i]=(e22626[i]/e22626cb[i])*60;
    if e22626c[i]=2 and e22626cb[i]>0 then hrwg[i]=e22626[i]/e22626cb[i];
    if e22626c[i]=3 and e22626cb[i]>0 and e22626e[i]>=0 and e22610[i]>0 then
      hrwg[i]=(e22626[i]/e22626cb[i])*e22626e[i]/e22610[i];
    if e22626c[i]=4 and e22626cb[i]>0 and e22610[i]>0 then hrwg[i]=(e22626[i]/e22626cb[i])/e22610[i];
    if e22626c[i]=5 and e22626cb[i]>0 and e22610[i]>0 then hrwg[i]=(e22626[i]/e22626cb[i])/(e22610[i]*4.3);
    /* missing value */
    if -4<e22626c[i]<0 or -4<e22626cb[i]<=0 then hrwg[i]=-3;
    if e22626c[i] in (3,4,5) and -4<e22610[i]<=0 then hrwg[i]=-3;
    if e22626c[i]=3 and e22626e[i]<0 then hrwg[i]=-3;
  end;
  if e22609[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e22626[i]<0 then
    hrwg[i]=e22626[i];
  if e22609[i] in (2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e22610[i]=0 then hrwg[i]=-3;
  if e22609[i] in (2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e22610[i]<0 then
    hrwg[i]=e22610[i];
end;
end;

do i=1 to 9;
  if e102251[i]=1 and e102252[i]=0 then do; /* rate incorrect but hours correct*/
    /* no overtime, use e98402 for hours */
    if e100230[i]=1 and e100248[i]>=0 then hrwg[i]=e100248[i];
    if e100230[i]=2 and e98402[i]>0 and e100248e[i]>0 and e100248[i]>=0 then
      hrwg[i]=e100248[i]*e100248e[i]/e98402[i];
    if e100230[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e98402[i]>0 and e100248[i]>=0
      then hrwg[i]=e100248[i]/e98402[i];
    if e100230[i]=4 and e98402[i]>0 and e100248[i]>=0 then hrwg[i]=e100248[i]/(2*e98402[i]);
    if e100230[i]=5 and e98402[i]>0 and e100248[i]>=0 then hrwg[i]=e100248[i]/(4.3*e98402[i]);
    if e100230[i]=6 and e98402[i]>0 and e99500[i]>0 and e100248[i]>=0 then
      hrwg[i]=e100248[i]/(e99500[i]*e98402[i]);
    if e100230[i]=8 and e98402[i]>0 and e100248[i]>=0 then hrwg[i]=e100248[i]/(2.15*e98402[i]);
    if e100230[i] in (9,14) then hrwg[i]=0;
    if e100230[i]=2 and e100248e[i] le 0 then hrwg[i]=-3;
    if e100230[i]=2 and -4<e100248e[i]<0 then hrwg[i]=e98402d[i];
    if e100230[i]=6 and e99500[i] le 0 then hrwg[i]=-3;
    if e100230[i]=6 and -4<e99500[i]<0 then hrwg[i]=e99500[i];
    if e100230[i] in (12,13) and e100248[i]>=0 then do;

```

Appendix 2: Employment Variable Creation

```

/* condition based on the time unit to finish per hrwg. */
if e100248c[i]=1 and e100248cb[i]>0 then hrwg[i]=(e100248[i]/e100248cb[i])*60;
if e100248c[i]=2 and e100248cb[i]>0 then hrwg[i]=e100248[i]/e100248cb[i];
if e100248c[i]=3 and e100248cb[i]>0 and e100248e[i]>=0 and e98402[i]>0 then
    hrwg[i]=(e100248[i]/e100248cb[i])*e100248e[i]/e98402[i];
if e100248c[i]=4 and e100248cb[i]>0 and e98402[i]>0 then hrwg[i]=(e100248[i]/e100248cb[i])/e98402[i];
if e100248c[i]=5 and e100248cb[i]>0 and e98402[i]>0 then hrwg[i]=(e100248[i]/e100248cb[i])/(e98402[i]*4.3);
/* missing value */
if -4<e100248c[i]<0 or -4<e100248cb[i]<=0 then hrwg[i]=-3;
if e100248c[i] in (3,4,5) and -4<e98402[i]<=0 then hrwg[i]=-3;
if e100248c[i]=3 and e100248e[i]<0 then hrwg[i]=-3;
end;

if e100230[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e100248[i]<0 then
    hrwg[i]=e100248[i];
if e100230[i] in (2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e98402[i]=0 then hrwg[i]=-3;
if e100230[i] in (2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e98402[i]<0 then
    hrwg[i]=e98402[i];

/* overtime, use e98429 for hours*/
if e100230[i]=1 and e100248[i]>=0 then hrwg[i]=e100248[i];
if e100230[i]=2 and e98429[i]>0 and e100248e[i]>0 and e100248[i]>=0 then
    hrwg[i]=e100248[i]*e100248e[i]/e98429[i];
if e100230[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999,-1,-2) and e98429[i]>0 and e100248[i]>=0
    then hrwg[i]=e100248[i]/e98429[i];
if e100230[i]=4 and e98429[i]>0 and e100248[i]>=0 then hrwg[i]=e100248[i]/(2*e98429[i]);
if e100230[i]=5 and e98429[i]>0 and e100248[i]>=0 then hrwg[i]=e100248[i]/(4.3*e98429[i]);
if e100230[i]=6 and e98429[i]>0 and e99500[i]>0 and e100248[i]>=0 then
    hrwg[i]=e100248[i]/(e99500[i]*e98429[i]);
if e100230[i]=8 and e98429[i]>0 and e100248[i]>=0 then hrwg[i]=e100248[i]/(2.15*e98429[i]);
if e100230[i] in (12,13) and e100248[i]>=0 then do;
    /* condition based on the time unit to finish per hrwg. */
    if e100248c[i]=3 and e100248cb[i]>0 and e100248e[i]>=0 and e98429[i]>0 then
        hrwg[i]=(e100248[i]/e100248cb[i])*e100248e[i]/e98429[i];
    if e100248c[i]=4 and e100248cb[i]>0 and e98429[i]>0 then hrwg[i]=(e100248[i]/e100248cb[i])/e98429[i];
    if e100248c[i]=5 and e100248cb[i]>0 and e98429[i]>0 then hrwg[i]=(e100248[i]/e100248cb[i])/(e98429[i]*4.3);
/* missing value */
    if e100248c[i] in (3,4,5) and -4<e98429[i]<=0 then hrwg[i]=-3;
end;

if e100230[i] in (2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e98429[i]=0 then hrwg[i]=-3;
if e100230[i] in (2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e98429[i]<0 then
    hrwg[i]=e98429[i];
end;

if e102251[i]=0 and e102252[i]=1 then do; /*rate correct but hours incorrect*/
    if e83100[i] ne 1 and e100231[i]>0 and e98402[i]>0 and hrwg[i]>=0 then hrwg[i]=hrwg[i]*e98402[i]/e100231[i];
    if e83100[i] ne 1 and e100231[i]>0 and e98429[i]>0 and hrwg[i]>=0 then hrwg[i]=hrwg[i]*e98429[i]/e100231[i];
    if e83100[i] ne 1 and e100231[i]=0 then hrwg[i]=-3;
    if e83100[i] ne 1 and -4<e100231[i]<0 then hrwg[i]=e100231[i];
end;

if e102251[i]=1 and e102252[i]=1 then do; /* neither rate nor hours is correct*/
    if e100230[i]=1 and e100248[i]>=0 then hrwg[i]=e100248[i];

```

Appendix 2: Employment Variable Creation

```

if e100230[i]=2 and e100231[i]>0 and e100248e[i]>0 and e100248[i]>=0 then
    hrwg[i]=e100248[i]*e100248e[i]/e100231[i];
if e100230[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e100231[i]>0 and e100248[i]>=0
    then hrwg[i]=e100248[i]/e100231[i];
if e100230[i]=4 and e100231[i]>0 and e100248[i]>=0 then hrwg[i]=e100248[i]/(2*e100231[i]);
if e100230[i]=5 and e100231[i]>0 and e100248[i]>=0 then hrwg[i]=e100248[i]/(4.3*e100231[i]);
if e100230[i]=6 and e100231[i]>0 and e99500[i]>0 and e100248[i]>=0 then
    hrwg[i]=e100248[i]/(e99500[i]*e100231[i]);
if e100230[i]=8 and e100231[i]>0 and e100248[i]>=0 then hrwg[i]=e100248[i]/(2.15*e100231[i]);
if e100230[i] in (9,14) then hrwg[i]=0;
if e100230[i]=2 and e100248e[i] le 0 then hrwg[i]=-3;
if e100230[i]=2 and -4<e100248e[i]<0 then hrwg[i]=e100248e[i];
if e100230[i]=2 and -4<e100248e[i]<0 then hrwg[i]=e100248e[i];
if e100230[i]=6 and e99500[i] le 0 then hrwg[i]=-3;
if e100230[i]=6 and -4<e99500[i]<0 then hrwg[i]=e99500[i];
if e100230[i] in (12,13) and e100248[i]>=0 then do;
    /* condition based on the time unit to finish per hrwg. */
    if e100248c[i]=1 and e100248cb[i]>0 then hrwg[i]=(e100248[i]/e100248cb[i])*60;
    if e100248c[i]=2 and e100248cb[i]>0 then hrwg[i]=e100248[i]/e100248cb[i];
    if e100248c[i]=3 and e100248cb[i]>0 and e100248e[i]>=0 and e100231[i]>0 then
        hrwg[i]=(e100248[i]/e100248cb[i])*e100248e[i]/e100231[i];
    if e100248c[i]=4 and e100248cb[i]>0 and e100231[i]>0 then hrwg[i]=(e100248[i]/e100248cb[i])/e100231[i];
    if e100248c[i]=5 and e100248cb[i]>0 and e100231[i]>0 then hrwg[i]=(e100248[i]/e100248cb[i])/(e100231[i]*4.3);
    /* missing value */
    if -4<e100248c[i]<0 or -4<e100248cb[i]<=0 then hrwg[i]=-3;
    if e100248c[i] in (3,4,5) and -4<e100231[i]<=0 then hrwg[i]=-3;
    if e100248c[i]=3 and e100248e[i]<0 then hrwg[i]=-3;
end;
if e100230[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e100248[i]<0 then
    hrwg[i]=e100248[i];
if e100230[i] in (2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e100231[i]=0 then
    hrwg[i]=-3;
if e100230[i] in (2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e100231[i]<0 then
    hrwg[i]=e100231[i];
end;
end;

***** add the start compensation *****
***** amount paid of overtime *****
ot1=-4; ot2=-4; ot3=-4; ot4=-4; ot5=-4; ot6=-4; ot7=-4; ot8=-4; ot9=-4;
otf1=0; otf2=0; otf3=0; otf4=0; otf5=0; otf6=0; otf7=0; otf8=0; otf9=0;

multot=0;

array ot ot1-ot9;
array otf otf1-otf9;

/* if report hourly rate of pay */
do i=1 to 9;

if e24502[i]=0 then multot=multot+1;
if e24501[i]>0 and e24502[i]=1 and e24514[i]>=0 then ot[i]=e24514[i];
if e24501[i]>0 and e24502[i]=2 and e24514b[i]>0 and e24514[i]>=0 then ot[i]=e24514[i]*e24514b[i]/e24501[i];
if e24501[i]>0 and e24502[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e24514[i]>=0 then
    ot[i]=e24514[i]/e24501[i];
if e24501[i]>0 and e24502[i]=4 and e24514[i]>=0 then ot[i]=e24514[i]/(2*e24501[i]);
if e24501[i]>0 and e24502[i]=5 and e24514[i]>=0 then ot[i]=e24514[i]/(4.3*e24501[i]);

```

Appendix 2: Employment Variable Creation

```

if e24501[i]>0 and e24502[i]=6 and e24514[i]>=0 then ot[i]=-3; /* since the weeks per year is unknown.*/
if e24501[i]>0 and e24502[i]=8 and e24514[i]>=0 then ot[i]=e24514[i]/(2.15*e24501[i]);
if e24501[i]>0 and e24502[i]=10 and hrwg[i]>=0 then ot[i]=hrwg[i]*1.5;
if e24501[i]>0 and e24502[i]=11 and hrwg[i]>=0 then ot[i]=hrwg[i]*2;
if e24502[i] in (9, 14) then ot[i]=0;

if -4<e24501[i]<0 then ot[i]=e24501[i];
if e24514[i]>=0 and e24501[i]=0 then ot[i]=-3;

if e24502[i] in (12,13) and e24514[i]>=0 then do;
/* condition based on the time unit to finish per hrwg. */
  if e24514aab[i]=1 and e24514aabb[i]>0 then ot[i]=(e24514[i]/e24514aabb[i])*60;
  if e24514aab[i]=2 and e24514aabb[i]>0 then ot[i]=e24514[i]/e24514aabb[i];
  if e24514aab[i]=3 and e24514aabb[i]>0 and e24514b[i]>=0 and e24501[i]>0 then
    ot[i]=(e24514[i]/e24514aabb[i])*e24514b[i]/e24501[i];
  if e24514aab[i]=4 and e24514aabb[i]>0 and e24501[i]>0 then ot[i]=(e24514[i]/e24514aabb[i])/e24501[i];
  if e24514aab[i]=5 and e24514aabb[i]>0 and e24501[i]>0 then ot[i]=(e24514[i]/e24514aabb[i])/(e24501[i]*4.3);
/* missing value */
  if -4<e24514aab[i]<0 or -4<e24514aabb[i]<=0 then ot[i]=-3;
  if e24514aab[i] in (3,4,5) and -4<e24501[i]<=0 then ot[i]=-3;
  if e24514aab[i]=3 and e24514b[i]<0 then ot[i]=-3;
end;

/*missing value*/;
if -4<e24514[i]<0 then ot[i]=e24514[i];
if e24502[i]=2 and e24514b[i]=0 then ot[i]=-3;
if e24502[i]=2 and -4<e24514b[i]<0 then ot[i]=e24514b[i];

end;

/*if report payment in other units*/

do i=1 to 9;

if e34404[i]=0 then multot=multot+1;
if e34403[i]>0 and e34404[i]=1 and e34413c[i]>=0 then ot[i]=e34413c[i];
if e34403[i]>0 and e34404[i]=2 and e34413e[i]>0 and e34413c[i]>=0 then ot[i]=e34413c[i]*e34413e[i]/e34403[i];
if e34403[i]>0 and e34404[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e34413c[i]>=0 then
  ot[i]=e34413c[i]/e34403[i];
if e34403[i]>0 and e34404[i]=4 and e34413c[i]>=0 then ot[i]=e34413c[i]/(2*e34403[i]);
if e34403[i]>0 and e34404[i]=5 and e34413c[i]>=0 then ot[i]=e34413c[i]/(4.3*e34403[i]);
if e34403[i]>0 and e34404[i]=6 and e34413c[i]>=0 then ot[i]=-3; /* since there is no weeks per year. */
if e34403[i]>0 and e34404[i]=8 and e34413c[i]>=0 then ot[i]=e34413c[i]/(2.15*e34403[i]);
if e34403[i]>0 and e34404[i]=10 and hrwg[i]>=0 then ot[i]=hrwg[i]*1.5;
if e34403[i]>0 and e34404[i]=11 and hrwg[i]>=0 then ot[i]=hrwg[i]*2;
if e34404[i] in (9,14) then ot[i]=0;

if -4<e34403[i]<0 then ot[i]=e34403[i];
if e34413c[i]>=0 and e34403[i]=0 then ot[i]=-3;

if e34404[i] in (12,13) and e34413c[i]>=0 then do;
/* condition based on the time unit to finish per hrwg. */
  if e34413cc[i]=1 and e34413cd[i]>0 then ot[i]=(e34413c[i]/e34413cd[i])*60;
  if e34413cc[i]=2 and e34413cd[i]>0 then ot[i]=e34413c[i]/e34413cd[i];
  if e34413cc[i]=3 and e34413cd[i]>0 and e34413e[i]>=0 and e34403[i]>0 then
    ot[i]=(e34413c[i]/e34413cd[i])*e34413e[i]/e34403[i];
  if e34413cc[i]=4 and e34413cd[i]>0 and e34403[i]>0 then ot[i]=(e34413c[i]/e34413cd[i])/e34403[i];

```

```

if e34413cc[i]=5 and e34413cd[i]>0 and e34403[i]>0 then ot[i]=(e34413c[i]/e34413cd[i])/(e34403[i]*4.3);
/* missing value */
if -4<e34413cc[i]<0 or -4<e34413cd[i]<=0 then ot[i]=-3;
if e34413cc[i] in (3,4,5) and -4<e34403[i]<=0 then ot[i]=-3;
if e34413cc[i]=3 and e34413e[i]<0 then ot[i]=-3;
end;

/*missing value*/;
if -4<e34413c[i]<0 then ot[i]=e34413c[i];
if e34404[i]=2 and e34413e[i]=0 then ot[i]=-3;
if e34404[i]=2 and -4<e34413e[i]<0 then ot[i]=e34413e[i];

end;

/*if report overtime pay later*/

do i=1 to 9;

if e88502[i]=0 then multot=multot+1;
if e88501[i]>0 and e88502[i]=1 and e88512[i]>=0 then ot[i]=e88512[i];
if e88501[i]>0 and e88502[i]=2 and e88512b[i]>0 and e88512[i]>=0 then ot[i]=e88512[i]*e88512b[i]/e88501[i];
if e88501[i]>0 and e88502[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e88512[i]>=0 then
    ot[i]=e88512[i]/e88501[i];
if e88501[i]>0 and e88502[i]=4 and e88512[i]>=0 then ot[i]=e88512[i]/(2*e88501[i]);
if e88501[i]>0 and e88502[i]=5 and e88512[i]>=0 then ot[i]=e88512[i]/(4.3*e88501[i]);
if e88501[i]>0 and e88502[i]=6 and e88512[i]>=0 then ot[i]=-3; /* since there is no weeks per year. */
if e88501[i]>0 and e88502[i]=8 and e88512[i]>=0 then ot[i]=e88512[i]/(2.15*e88501[i]);
if e88501[i]>0 and e88502[i]=10 and hrwg[i]>=0 then ot[i]=hrwg[i]*1.5;
if e88501[i]>0 and e88502[i]=11 and hrwg[i]>=0 then ot[i]=hrwg[i]*2;
if e88502[i] in (9,14) then ot[i]=0;
if -4<e88501[i]<0 then ot[i]=e88501[i];
if e88512[i]>=0 and e88501[i]=0 then ot[i]=-3;
if e88502[i] in (12,13) and e88512[i]>=0 then do;
    /* condition based on the time unit to finish per hrwg. */
    if e88512aaa[i]=1 and e88512aab[i]>0 then ot[i]=(e88512[i]/e88512aab[i])*60;
    if e88512aaa[i]=2 and e88512aab[i]>0 then ot[i]=e88512[i]/e88512aab[i];
    if e88512aaa[i]=3 and e88512aab[i]>0 and e88512b[i]>=0 and e88501[i]>0 then
        ot[i]=(e88512[i]/e88512aab[i])*e88512b[i]/e88501[i];
    if e88512aaa[i]=4 and e88512aab[i]>0 and e88501[i]>0 then ot[i]=(e88512[i]/e88512aab[i])/e88501[i];
    if e88512aaa[i]=5 and e88512aab[i]>0 and e88501[i]>0 then ot[i]=(e88512[i]/e88512aab[i])/(e88501[i]*4.3);
    /* missing value */
    if -4<e88512aaa[i]<0 or -4<e88512aab[i]<=0 then ot[i]=-3;
    if e88512aaa[i] in (3,4,5) and -4<e88501[i]<=0 then ot[i]=-3;
    if e88512aaa[i]=3 and e88512b[i]<0 then ot[i]=-3;
end;

/*missing value*/;
if -4<e88512[i]<0 then ot[i]=e88512[i];
if e88502[i]=2 and e88512b[i]=0 then ot[i]=-3;
if e88502[i]=2 and -4<e88512b[i]<0 then ot[i]=e88512b[i];

end;

do i=1 to 9;

if e98404a[i]=0 then multot=multot+1;
if e98403[i]>0 and e98404a[i]=1 and e98414[i]>=0 then ot[i]=e98414[i];

```

Appendix 2: Employment Variable Creation

```

if e98403[i]>0 and e98404a[i]=2 and e98414b[i]>0 and e98414[i]>=0 then ot[i]=e98414[i]*e98414b[i]/e98403[i];
if e98403[i]>0 and e98404a[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e98414[i]>=0 then
    ot[i]=e98414[i]/e98403[i];
if e98403[i]>0 and e98404a[i]=4 and e98414[i]>=0 then ot[i]=e98414[i]/(2*e98403[i]);
if e98403[i]>0 and e98404a[i]=5 and e98414[i]>=0 then ot[i]=e98414[i]/(4.3*e98403[i]);
if e98403[i]>0 and e98404a[i]=6 and e98414[i]>=0 then ot[i]=-3; /* since there is no weeks per year. */
if e98403[i]>0 and e98404a[i]=8 and e98414[i]>=0 then ot[i]=e98414[i]/(2.15*e98403[i]);
if e98403[i]>0 and e98404a[i]=10 and hrgw[i]>=0 then ot[i]=hrgw[i]*1.5;
if e98403[i]>0 and e98404a[i]=11 and hrgw[i]>=0 then ot[i]=hrgw[i]*2;
if e98404a[i] in (9,14) then ot[i]=0;
if -4<e98403[i]<0 then ot[i]=e98403[i];
if e98414[i]>=0 and e98403[i]=0 then ot[i]=-3;
if e98404a[i] in (12,13) and e98414[i]>=0 then do;
    /* condition based on the time unit to finish per hrgw. */
    if e98414aaa[i]=1 and e98414aab[i]>0 then ot[i]=(e98414[i]/e98414aab[i])*60;
    if e98414aaa[i]=2 and e98414aab[i]>0 then ot[i]=e98414[i]/e98414aab[i];
    if e98414aaa[i]=3 and e98414aab[i]>0 and e98414b[i]>=0 and e98403[i]>0 then
        ot[i]=(e98414[i]/e98414aab[i])*e98414b[i]/e98403[i];
    if e98414aaa[i]=4 and e98414aab[i]>0 and e98403[i]>0 then ot[i]=(e98414[i]/e98414aab[i])/e98403[i];
    if e98414aaa[i]=5 and e98414aab[i]>0 and e98403[i]>0 then ot[i]=(e98414[i]/e98414aab[i])/(e98403[i]*4.3);
    /* missing value */
    if -4<e98414aaa[i]<0 or -4<e98414aab[i]<=0 then ot[i]=-3;
    if e98414aaa[i] in (3,4,5) and -4<e98403[i]<=0 then ot[i]=-3;
    if e98414aaa[i]=3 and e98414b[i]<0 then ot[i]=-3;
end;

/*missing value*/;
if -4<e98414[i]<0 then ot[i]=e98414[i];
if e98404a[i]=2 and e98414b[i]=0 then ot[i]=-3;
if e98404a[i]=2 and -4<e98414b[i]<0 then ot[i]=e98414b[i];

end;

***** report the corrected overtime payment if the correction is made *****/
do i=1 to 9;

if e226044[i]=1 and e226043[i]=0 then do; /*rate incorrect but hours correct*/
    if e19200[i]=1 then do; /* report hourly payrate*/
        if e22612[i]=0 then multot=multot+1;
        if e22612[i]=1 and e22627[i]>=0 then ot[i]=e22627[i];
        if e22612[i]=2 and e22627[i]>=0 and e22627e[i]>0 and e24501[i]>0 then ot[i]=e22627[i]*e22627e[i]/e24501[i];
        if e22612[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e22627[i]>=0 and e24501[i]>0
            then ot[i]=e22627[i]/e24501[i];
        if e22612[i]=4 and e22627[i]>=0 and e24501[i]>0 then ot[i]=e22627[i]/(2*e24501[i]);
        if e22612[i]=5 and e22627[i]>=0 and e24501[i]>0 then ot[i]=e22627[i]/(4.3*e24501[i]);
        if e22612[i]=6 and e22627[i]>=0 and e24501[i]>0 then ot[i]=-3; /* since there is no weeks per year. */
        if e22612[i]=8 and e22627[i]>=0 and e24501[i]>0 then ot[i]=e22627[i]/(2.15*e24501[i]);
        if e22612[i] in (9,14) then ot[i]=0;
        if e22612[i] in (12,13) and e22627[i]>=0 then do;
            /* condition based on the time unit to finish per ot. */
            if e22627c[i]=1 and e22627cb[i]>0 then ot[i]=(e22627[i]/e22627cb[i])*60;
            if e22627c[i]=2 and e22627cb[i]>0 then ot[i]=e22627[i]/e22627cb[i];
            if e22627c[i]=3 and e22627cb[i]>0 and e22627e[i]>0 and e24501[i]>0 then
                ot[i]=(e22627[i]/e22627cb[i])*e22627e[i]/e24501[i];
            if e22627c[i]=4 and e22627cb[i]>0 and e24501[i]>0 then ot[i]=(e22627[i]/e22627cb[i])/e24501[i];
            if e22627c[i]=5 and e22627cb[i]>0 and e24501[i]>0 then ot[i]=(e22627[i]/e22627cb[i])/(e24501[i]*4.3);
            /* missing value */

```

Appendix 2: Employment Variable Creation

```

if -4<e22627c[i]<0 or -4<e22627cb[i]<=0 then ot[i]=-3;
  if e22627c[i] in (3,4,5) and -4<e24501[i]<=0 then ot[i]=-3;
    if e22627c[i]=3 and e22627e[i]<0 then ot[i]=-3;
  end;
if e22612[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e22627[i]<0 then
  ot[i]=e22627[i];
if e22612[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e24501[i]=0 then ot[i]=-3;
if e22612[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e24501[i]<0 then
  ot[i]=e24501[i];
if e22612[i]=2 and e22627e[i] le 0 then ot[i]=-3;
  if e22612[i]=2 and -4<e22627e[i]<0 then ot[i]=e22627e[i];
end;

if e19200[i] ne 1 then do; /*report non-hourly payrate*/
  if e22612[i]=0 then multot=multot+1;
  if e22612[i]=1 and e22627[i]>=0 then ot[i]=e22627[i];
  if e22612[i]=2 and e22627[i]>=0 and e22627e[i]>0 and e34403[i]>0 then ot[i]=e22627[i]*e22627e[i]/e34403[i];
  if e22612[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999,-1,-2) and e22627[i]>=0 and e34403[i]>0
    then ot[i]=e22627[i]/e34403[i];
  if e22612[i]=4 and e22627[i]>=0 and e34403[i]>0 then ot[i]=e22627[i]/(2*e34403[i]);
  if e22612[i]=5 and e22627[i]>=0 and e34403[i]>0 then ot[i]=e22627[i]/(4.3*e34403[i]);
  if e22612[i]=6 and e22627[i]>=0 and e34403[i]>0 then ot[i]=-3; /* since there is no weeks per year.*/
  if e22612[i]=8 and e22627[i]>=0 and e34403[i]>0 then ot[i]=e22627[i]/(2.15*e34403[i]);
  if e22612[i] in (9,14) then ot[i]=0;
  if e22612[i] in (12,13) and e22627[i]>=0 then do;
    /* condition based on the time unit to finish per ot. */
    if e22627c[i]=1 and e22627cb[i]>0 then ot[i]=(e22627[i]/e22627cb[i])*60;
    if e22627c[i]=2 and e22627cb[i]>0 then ot[i]=e22627[i]/e22627cb[i];
    if e22627c[i]=3 and e22627cb[i]>0 and e22627e[i]>=0 and e34403[i]>0 then
      ot[i]=(e22627[i]/e22627cb[i])*e22627e[i]/e34403[i];
    if e22627c[i]=4 and e22627cb[i]>0 and e34403[i]>0 then ot[i]=(e22627[i]/e22627cb[i])/e34403[i];
    if e22627c[i]=5 and e22627cb[i]>0 and e34403[i]>0 then ot[i]=(e22627[i]/e22627cb[i])/(e34403[i]*4.3);
  /* missing value */
    if -4<e22627c[i]<0 or -4<e22627cb[i]<=0 then ot[i]=-3;
    if e22627c[i] in (3,4,5) and -4<e34403[i]<=0 then ot[i]=-3;
      if e22627c[i]=3 and e22627e[i]<0 then ot[i]=-3;
    end;
  if e22612[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e22627[i]<0 then
    ot[i]=e22627[i];
  if e22612[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34403[i]=0 then ot[i]=-3;
  if e22612[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34403[i]<0 then
    ot[i]=e34403[i];
  if e22612[i]=2 and e22627e[i] le 0 then ot[i]=-3;
  if e22612[i]=2 and -4<e22627e[i]<0 then ot[i]=e22627e[i];
end;
end;

if e226044[i]=0 and e226043[i]=1 then do; /*rate correct but hours incorrect*/
  if e24502[i] ne 1 and ot[i]>=0 and e34403[i]>0 and e22611[i]>0 then ot[i]=ot[i]*e34403[i]/e22611[i];
  if e34404[i] ne 1 and ot[i]>=0 and e34403[i]>0 and e22611[i]>0 then ot[i]=ot[i]*e34403[i]/e22611[i];
  if e24502[i] ne 1 and e34404[i] ne 1 and e22611[i]=0 then ot[i]=-3;
  if e24502[i] ne 1 and e34404[i] ne 1 and -4<e22611[i]<0 then ot[i]=e22611[i];
end;

if e226044[i]=1 and e226043[i]=1 then do; /* neither rate nor the hours is correct*/
  if e22612[i]=0 then multot=multot+1;
  if e22612[i]=1 and e22627[i]>=0 then ot[i]=e22627[i];

```

Appendix 2: Employment Variable Creation

```

if e22612[i]=2 and e22627[i]>=0 and e22627e[i]>0 and e22611[i]>0 then ot[i]=e22627[i]*e22627e[i]/e22611[i];
if e22612[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999,-1,-2) and e22627[i]>=0 and e22611[i]>0 then
    ot[i]=e22627[i]/e22611[i];
if e22612[i]=4 and e22627[i]>=0 and e22611[i]>0 then ot[i]=e22627[i]/(2*e22611[i]);
if e22612[i]=5 and e22627[i]>=0 and e22611[i]>0 then ot[i]=e22627[i]/(4.3*e22611[i]);
if e22612[i]=6 and e22627[i]>=0 and e22611[i]>0 then ot[i]=-3; /* since there is no weeks per year. */
if e22612[i]=8 and e22627[i]>=0 and e22611[i]>0 then ot[i]=e22627[i]/(2.15*e22611[i]);
if e22612[i] in (9,14) then ot[i]=0;
if e22612[i] in (12,13) and e22627[i]>=0 then do;
    /* condition based on the time unit to finish per ot. */
    if e22627c[i]=1 and e22627cb[i]>0 then ot[i]=(e22627[i]/e22627cb[i])*60;
    if e22627c[i]=2 and e22627cb[i]>0 then ot[i]=e22627[i]/e22627cb[i];
    if e22627c[i]=3 and e22627cb[i]>0 and e22627e[i]>=0 and e22611[i]>0 then
        ot[i]=(e22627[i]/e22627cb[i])*e22627e[i]/e22611[i];
    if e22627c[i]=4 and e22627cb[i]>0 and e22611[i]>0 then ot[i]=(e22627[i]/e22627cb[i])/e22611[i];
    if e22627c[i]=5 and e22627cb[i]>0 and e22611[i]>0 then ot[i]=(e22627[i]/e22627cb[i])/(e22611[i]*4.3);
/* missing value */
    if -4<e22627c[i]<0 or -4<e22627cb[i]<=0 then ot[i]=-3;
    if e22627c[i] in (3,4,5) and -4<e22611[i]<=0 then ot[i]=-3;
    if e22627c[i]=3 and e22627e[i]<0 then ot[i]=-3;
end;
if e22612[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e22627[i]<0 then
    ot[i]=e22627[i];
if e22612[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e22611[i]=0 then ot[i]=-3;
if e22612[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e22611[i]<0 then
    ot[i]=e22611[i];
if e22612[i]=2 and e22627e[i] le 0 then ot[i]=-3;
if e22612[i]=2 and -4<e22627e[i]<0 then ot[i]=e22627e[i];

end;
end;

***** correction made in the later part *****

do i=1 to 9;

if e102254[i]=1 and e102253[i]=0 then do; /*rate incorrect but hours correct*/
    if e83100[i]=1 then do; /* report hourly payrate*/
        if e100233[i]=0 then multot=multot+1;
        if e100233[i]=1 and e100249[i]>=0 then ot[i]=e100249[i];
        if e100233[i]=2 and e100249[i]>=0 and e100249e[i]>0 and e88501[i]>0 then
            ot[i]=e100249[i]*e100249e[i]/e88501[i];
        if e100233[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999,-1,-2) and e100249[i]>=0 and e88501[i]>0
            then ot[i]=e100249[i]/e88501[i];
        if e100233[i]=4 and e100249[i]>=0 and e88501[i]>0 then ot[i]=e100249[i]/(2*e88501[i]);
        if e100233[i]=5 and e100249[i]>=0 and e88501[i]>0 then ot[i]=e100249[i]/(4.3*e88501[i]);
        if e100233[i]=6 and e100249[i]>=0 and e88501[i]>0 then ot[i]=-3; /* since there is no weeks per year. */
        if e100233[i]=8 and e100249[i]>=0 and e88501[i]>0 then ot[i]=e100249[i]/(2.15*e88501[i]);
        if e100233[i] in (9,14) then ot[i]=0;
        if e100233[i] in (12,13) and e100249[i]>=0 then do;
            /* condition based on the time unit to finish per ot. */
            if e100249c[i]=1 and e100249cb[i]>0 then ot[i]=(e100249[i]/e100249cb[i])*60;
            if e100249c[i]=2 and e100249cb[i]>0 then ot[i]=e100249[i]/e100249cb[i];
            if e100249c[i]=3 and e100249cb[i]>0 and e100249e[i]>=0 and e88501[i]>0 then
                ot[i]=(e100249[i]/e100249cb[i])*e100249e[i]/e88501[i];
            if e100249c[i]=4 and e100249cb[i]>0 and e88501[i]>0 then ot[i]=(e100249[i]/e100249cb[i])/e88501[i];
            if e100249c[i]=5 and e100249cb[i]>0 and e88501[i]>0 then ot[i]=(e100249[i]/e100249cb[i])/(e88501[i]*4.3);

```

```

/* missing value */
if -4<e100249c[i]<0 or -4<e100249cb[i]<=0 then ot[i]=-3;
if e100249c[i] in (3,4,5) and -4<e88501[i]<=0 then ot[i]=-3;
if e100249c[i]=3 and e100249e[i]<0 then ot[i]=-3;
end;
if e100233[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e100249[i]<0 then
    ot[i]=e100249[i];
if e100233[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e88501[i]=0 then ot[i]=-3;
if e100233[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e88501[i]<0 then
    ot[i]=e88501[i];
if e100233[i]=2 and e100249e[i] le 0 then ot[i]=-3;
if e100233[i]=2 and -4<e100249e[i]<0 then ot[i]=e100249e[i];

end;

if e83100[i] ne 1 then do; /*report non-hourly payrate*/
    if e100233[i]=0 then multot=multot+1;
    if e100233[i]=1 and e100249[i]>=0 then ot[i]=e100249[i];
    if e100233[i]=2 and e100249[i]>=0 and e100249e[i]>0 and e98403[i]>0 then
        ot[i]=e100249[i]*e100249e[i]/e98403[i];
    if e100233[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999,-1,-2) and e100249[i]>=0 and e98403[i]>0
        then ot[i]=e100249[i]/e98403[i];
    if e100233[i]=4 and e100249[i]>=0 and e98403[i]>0 then ot[i]=e100249[i]/(2*e98403[i]);
    if e100233[i]=5 and e100249[i]>=0 and e98403[i]>0 then ot[i]=e100249[i]/(4.3*e98403[i]);
    if e100233[i]=6 and e100249[i]>=0 and e98403[i]>0 then ot[i]=-3; /* since there is no weeks per year.*/
    if e100233[i]=8 and e100249[i]>=0 and e98403[i]>0 then ot[i]=e100249[i]/(2.15*e98403[i]);
    if e100233[i] in (9,14) then ot[i]=0;
    if e100233[i] in (12,13) and e100249[i]>=0 then do;
        /* condition based on the time unit to finish per ot. */
        if e100249c[i]=1 and e100249cb[i]>0 then ot[i]=(e100249[i]/e100249cb[i])*60;
        if e100249c[i]=2 and e100249cb[i]>0 then ot[i]=e100249[i]/e100249cb[i];
        if e100249c[i]=3 and e100249cb[i]>0 and e100249e[i]>0 and e98403[i]>0 then
            ot[i]=(e100249[i]/e100249cb[i])*e100249e[i]/e98403[i];
        if e100249c[i]=4 and e100249cb[i]>0 and e98403[i]>0 then ot[i]=(e100249[i]/e100249cb[i])/e98403[i];
        if e100249c[i]=5 and e100249cb[i]>0 and e98403[i]>0 then ot[i]=(e100249[i]/e100249cb[i])/(e98403[i]*4.3);
        /* missing value */
        if -4<e100249c[i]<0 or -4<e100249cb[i]<=0 then ot[i]=-3;
        if e100249c[i] in (3,4,5) and -4<e98403[i]<=0 then ot[i]=-3;
        if e100249c[i]=3 and e100249e[i]<0 then ot[i]=-3;
    end;
    if e100233[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e100249[i]<0 then
        ot[i]=e100249[i];
    if e100233[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e98403[i]=0 then ot[i]=-3;
    if e100233[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e98403[i]<0 then
        ot[i]=e98403[i];
    if e100233[i]=2 and e100249e[i] le 0 then ot[i]=-3;
    if e100233[i]=2 and -4<e100249e[i]<0 then ot[i]=e100249e[i];
    end;
    end;

if e102254[i]=0 and e102253[i]=1 then do; /*rate correct but hours incorrect*/
    if e88502[i] ne 1 and ot[i]>=0 and e88501[i]>0 and e100232[i]>0 then ot[i]=ot[i]*e88501[i]/e100232[i];
    if e98404a[i] ne 1 and ot[i]>=0 and e98403[i]>0 and e100232[i]>0 then ot[i]=ot[i]*e98403[i]/e100232[i];
    if e88502[i] ne 1 and e98404a[i] ne 1 and e100232[i]=0 then ot[i]=-3;
    if e88502[i] ne 1 and e98404a[i] ne 1 and -4<e100232[i]<0 then ot[i]=e100232[i];
end;

```

```

if e102254[i]=1 and e102253[i]=1 then do; /* neither rate nor the hours is correct*/
  if e100233[i]=0 then multot=multot+1;
  if e100233[i]=1 and e100249[i]>=0 then ot[i]=e100249[i];
  if e100233[i]=2 and e100249[i]>=0 and e100249e[i]>0 and e100232[i]>0 then
    ot[i]=e100249[i]*e100249e[i]/e100232[i];
  if e100233[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999,-1,-2) and e100249[i]>=0 and e100232[i]>0
    then ot[i]=e100249[i]/e100232[i];
  if e100233[i]=4 and e100249[i]>=0 and e100232[i]>0 then ot[i]=e100249[i]/(2*e100232[i]);
  if e100233[i]=5 and e100249[i]>=0 and e100232[i]>0 then ot[i]=e100249[i]/(4.3*e100232[i]);
  if e100233[i]=6 and e100249[i]>=0 and e100232[i]>0 then ot[i]=-3; /* since there is no weeks per year. */
  if e100233[i]=8 and e100249[i]>=0 and e100232[i]>0 then ot[i]=e100249[i]/(2.15*e100232[i]);
  if e100233[i] in (9,14) then ot[i]=0;
  if e100233[i] in (12,13) and e100249[i]>=0 then do;
    /* condition based on the time unit to finish per ot. */
    if e100249c[i]=1 and e100249cb[i]>0 then ot[i]=(e100249[i]/e100249cb[i])*60;
    if e100249c[i]=2 and e100249cb[i]>0 then ot[i]=e100249[i]/e100249cb[i];
    if e100249c[i]=3 and e100249cb[i]>0 and e100249e[i]>0 and e100232[i]>0 then
      ot[i]=(e100249[i]/e100249cb[i])*e100249e[i]/e100232[i];
    if e100249c[i]=4 and e100249cb[i]>0 and e100232[i]>0 then ot[i]=(e100249[i]/e100249cb[i])/e100232[i];
    if e100249c[i]=5 and e100249cb[i]>0 and e100232[i]>0 then ot[i]=(e100249[i]/e100249cb[i])/(e100232[i]*4.3);
    /* missing value */
    if -4<e100249c[i]<0 or -4<e100249cb[i]<=0 then ot[i]=-3;
    if e100249c[i] in (3,4,5) and -4<e100232[i]<=0 then ot[i]=-3;
    if e100249c[i]=3 and e100249e[i]<0 then ot[i]=-3;
  end;
  if e100233[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e100249[i]<0 then
    ot[i]=e100249[i];
  if e100233[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e100232[i]=0 then ot[i]=-3;
  if e100233[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e100232[i]<0 then
    ot[i]=e100232[i];
  if e100233[i]=2 and e100249e[i] le 0 then ot[i]=-3;
  if e100233[i]=2 and -4<e100249e[i]<0 then ot[i]=e100249e[i];
end;
end;

/* amount paid on tips, commissions, bonuses, incentive pay or other*/
array othpay1 othpay11 othpay21 othpay31 othpay41 othpay51 othpay61 othpay71 othpay81 othpay91;
array othpay2 othpay12 othpay22 othpay32 othpay42 othpay52 othpay62 othpay72 othpay82 othpay92;
array othpay3 othpay13 othpay23 othpay33 othpay43 othpay53 othpay63 othpay73 othpay83 othpay93;
array othpay4 othpay14 othpay24 othpay34 othpay44 othpay54 othpay64 othpay74 othpay84 othpay94;
array othpay5 othpay15 othpay25 othpay35 othpay45 othpay55 othpay65 othpay75 othpay85 othpay95;

array othpf1 othpf11 othpf21 othpf31 othpf41 othpf51 othpf61 othpf71 othpf81 othpf91;
array othpf2 othpf12 othpf22 othpf32 othpf42 othpf52 othpf62 othpf72 othpf82 othpf92;
array othpf3 othpf13 othpf23 othpf33 othpf43 othpf53 othpf63 othpf73 othpf83 othpf93;
array othpf4 othpf14 othpf24 othpf34 othpf44 othpf54 othpf64 othpf74 othpf84 othpf94;
array othpf5 othpf15 othpf25 othpf35 othpf45 othpf55 othpf65 othpf75 othpf85 othpf95;

do i=1 to 9;
  othpay1[i]=-4;  othpay2[i]=-4;  othpay3[i]=-4;  othpay4[i]=-4;  othpay5[i]=-4;
  othpf1[i]=0;   othpf2[i]=0;   othpf3[i]=0;   othpf4[i]=0;   othpf5[i]=0;
end;

/* do it for non-overtime payment*/
***** for tips *****/

```

```

/*report hourly wage at the beginning*/
do i=1 to 9;
if e23901[i]>0 and e216001[i]=1 and e225001[i]>=0 then othpay1[i]=e225001[i];
if e23901[i]>0 and e216001[i]=2 and e225501[i] and e225001[i]>=0 then
    othpay1[i]=(e225001[i]*e225501[i])/e23901[i];
if e23901[i]>0 and e216001[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e225001[i]>=0
    then othpay1[i]=e225001[i]/e23901[i];
if e23901[i]>0 and e216001[i]=4 and e225001[i]>=0 then othpay1[i]=e225001[i]/(2*e23901[i]);
if e23901[i]>0 and e216001[i]=5 and e225001[i]>=0 then othpay1[i]=e225001[i]/(4.3*e23901[i]);
if e23901[i]>0 and e216001[i]=6 and e225001[i]>=0 and e35600[i]>0 then othpay1[i]=e225001[i]/(e35600[i]
    *e23901[i]);
if e23901[i]>0 and e216001[i]=8 and e225001[i]>=0 then othpay1[i]=e225001[i]/(2.15*e23901[i]);
if e216001[i] in (9,14) then othpay1[i]=0;
if e216001[i] in (12,13) and e225001[i]>=0 then do;
    /* condition based on the time unit to finish per othpay1. */
    if e225301[i]=1 and e225351[i]>0 then othpay1[i]=(e225001[i]/e225351[i])*60;
    if e225301[i]=2 and e225351[i]>0 then othpay1[i]=e225001[i]/e225351[i];
    if e225301[i]=3 and e225351[i]>0 and e225501[i]>=0 and e23901[i]>0 then
        othpay1[i]=(e225001[i]/e225351[i])*e225501[i]/e23901[i];
    if e225301[i]=4 and e225351[i]>0 and e23901[i]>0 then othpay1[i]=(e225001[i]/e225351[i])/e23901[i];
    if e225301[i]=5 and e225351[i]>0 and e23901[i]>0 then othpay1[i]=(e225001[i]/e225351[i])/(e23901[i]*4.3);
    /* missing value */
    if -4<e225301[i]<0 or -4<e225351[i]<=0 then othpay1[i]=-3;
    if e225301[i] in (3,4,5) and -4<e23901[i]<=0 then othpay1[i]=-3;
    if e225301[i]=3 and e225501[i]<0 then othpay1[i]=-3;
end;
if e216001[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999, -1, -2) and -4<e225001[i]<0 then
    othpay1[i]=e225001[i];
if e216001[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999, -1, -2) and e23901[i]=0 then othpay1[i]=-3;
if e216001[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999, -1, -2) and -4<e23901[i]<0 then
    othpay1[i]=e23901[i];
if e216001[i]=2 and e225501[i] le 0 then othpay1[i]=-3;
if e216001[i]=2 and -4<e225501[i]<0 then othpay1[i]=e225501[i];
end;

/*non-hourly wage at the beginning, with overtime*/
do i=1 to 9;
if e34428[i]>0 and e216001[i]=1 and e225001[i]>=0 then othpay1[i]=e225001[i];
if e34428[i]>0 and e216001[i]=2 and e34430[i]>0 and e225001[i]>=0 then
    othpay1[i]=(e225001[i]*e34430[i])/e34428[i];
if e34428[i]>0 and e216001[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e225001[i]>=0
    then othpay1[i]=e225001[i]/e34428[i];
if e34428[i]>0 and e216001[i]=4 and e225001[i]>=0 then othpay1[i]=e225001[i]/(2*e34428[i]);
if e34428[i]>0 and e216001[i]=5 and e225001[i]>=0 then othpay1[i]=e225001[i]/(4.3*e34428[i]);
if e34428[i]>0 and e216001[i]=6 and e225001[i]>=0 and e35600[i]>0 then othpay1[i]=e225001[i]/(e35600[i]
    *e34428[i]);
if e34428[i]>0 and e216001[i]=8 and e225001[i]>=0 then othpay1[i]=e225001[i]/(2.15*e34428[i]);
if e216001[i] in (12,13) and e225001[i]>=0 then do;
    /* condition based on the time unit to finish per othpay1. */
    if e225301[i]=3 and e225351[i]>0 and e225501[i]>=0 and e34428[i]>0 then
        othpay1[i]=(e225001[i]/e225351[i])*e225501[i]/e34428[i];
    if e225301[i]=4 and e225351[i]>0 and e34428[i]>0 then othpay1[i]=(e225001[i]/e225351[i])/e34428[i];
    if e225301[i]=5 and e225351[i]>0 and e34428[i]>0 then othpay1[i]=(e225001[i]/e225351[i])/(e34428[i]*4.3);
    /* missing value */
    if e225301[i] in (3,4,5) and -4<e34428[i]<=0 then othpay1[i]=-3;

```

Appendix 2: Employment Variable Creation

```

end;
if e216001[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34428[i]=0 then othpay1[i]=-3;
if e216001[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34428[i]<0 then
    othpay1[i]=e34428[i];
end;

/* non-hourly wage at the beginning, without overtime*/
do i=1 to 9;
if e34402[i]>0 and e216001[i]=1 and e225001[i]>=0 then othpay1[i]=e225001[i];
if e34402[i]>0 and e216001[i]=2 and e225501[i]>0 and e225001[i]>=0 then
    othpay1[i]=(e225001[i]*e225501[i])/e34402[i];
if e34402[i]>0 and e216001[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e225001[i]>=0
    then othpay1[i]=e225001[i]/e34402[i];
if e34402[i]>0 and e216001[i]=4 and e225001[i]>=0 then othpay1[i]=e225001[i]/(2*e34402[i]);
if e34402[i]>0 and e216001[i]=5 and e225001[i]>=0 then othpay1[i]=e225001[i]/(4.3*e34402[i]);
if e34402[i]>0 and e216001[i]=6 and e225001[i]>=0 and e35600[i]>0 then
    othpay1[i]=e225001[i]/(e35600[i]*e34402[i]);
if e34402[i]>0 and e216001[i]=8 and e225001[i]>=0 then othpay1[i]=e225001[i]/(2.15*e34402[i]);
if e216001[i] in (7,9,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then othpf1[i]=othpf1[i]+1;
if e216001[i] in (12,13) and e225001[i]>=0 then do;
    /* condition based on the time unit to finish per othpay1. */
    if e225301[i]=3 and e225351[i]>0 and e225501[i]>=0 and e34402[i]>0 then
        othpay1[i]=(e225001[i]/e225351[i])*e225501[i]/e34402[i];
    if e225301[i]=4 and e225351[i]>0 and e34402[i]>0 then othpay1[i]=(e225001[i]/e225351[i])/e34402[i];
    if e225301[i]=5 and e225351[i]>0 and e34402[i]>0 then othpay1[i]=(e225001[i]/e225351[i])/(e34402[i]*4.3);
/* missing value */
    if e225301[i] in (3,4,5) and -4<e34402[i]<=0 then othpay1[i]=-3;
end;
if e216001[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay1[i]=-3;
if e216001[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
    othpay1[i]=e34402[i];

if e216001[i]=6 and e35600[i] le 0 then othpay1[i]=-3;
if e216001[i]=6 and -4<e35600[i]<0 then othpay1[i]=e35600[i];
end;

/* if tips is corrected in the later part */
do i=1 to 9;
if e226045[i]=1 then do;
/*report hourly wage at the beginning*/
if e22613[i]=1 and e22628[i]>=0 then othpay1[i]=e22628[i];
if e22613[i]=2 and e22628[i]>=0 and e34430[i]>0 and e23901[i]>0 then othpay1[i]=e22628[i]*e34430[i]/e23901[i];
if e22613[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999,-1,-2) and e22628[i]>=0 and e23901[i]>0 then
    othpay1[i]=e22628[i]/e23901[i];
if e22613[i]=4 and e22628[i]>=0 and e23901[i]>0 then othpay1[i]=e22628[i]/(2*e23901[i]);
if e22613[i]=5 and e22628[i]>=0 and e23901[i]>0 then othpay1[i]=e22628[i]/(4.3*e23901[i]);
if e22613[i]=6 and e22628[i]>=0 and e23901[i]>0 and e35600[i]>0 then othpay1[i]=e22628[i]/(e35600[i]*e23901[i]);
if e22613[i]=8 and e22628[i]>=0 and e23901[i]>0 then othpay1[i]=e22628[i]/(2.15*e23901[i]);
if e22613[i] in (9,14) then othpay1[i]=0;
if e22613[i] in (12,13) and e22628[i]>=0 then do;
    /* condition based on the time unit to finish per othpay1. */
    if e22628c[i]=1 and e22628cb[i]>0 then othpay1[i]=(e22628[i]/e22628cb[i])*60;
    if e22628c[i]=2 and e22628cb[i]>0 then othpay1[i]=e22628[i]/e22628cb[i];
    if e22628c[i]=3 and e22628cb[i]>0 and e22628e[i]>=0 and e23901[i]>0 then
        othpay1[i]=(e22628[i]/e22628cb[i])*e22628e[i]/e23901[i];
    if e22628c[i]=4 and e22628cb[i]>0 and e23901[i]>0 then othpay1[i]=(e22628[i]/e22628cb[i])/e23901[i];
    if e22628c[i]=5 and e22628cb[i]>0 and e23901[i]>0 then othpay1[i]=(e22628[i]/e22628cb[i])/(e23901[i]*4.3);

```

```

/* missing value */
if -4<e22628c[i]<0 or -4<e22628cb[i]<=0 then othpay1[i]=-3;
if e22628c[i] in (3,4,5) and -4<e23901[i]<=0 then othpay1[i]=-3;
if e22628c[i]=3 and e22628e[i]<0 then othpay1[i]=-3;
end;
if e22613[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e22628[i]<0 then
    othpay1[i]=e22628[i];
if e22613[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay1[i]=-3;
if e22613[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay1[i]=e23901[i];
if e22613[i]=2 and e22628e[i] le 0 then othpay1[i]=-3;
if e22613[i]=2 and -4<e22628e[i]<0 then othpay1[i]=e22628e[i];

/* non-hourly wage at the beginning, with overtime*/
if e22613[i]=2 and e22628[i]>=0 and e22628e[i]>0 and e34428[i]>0 then othpay1[i]=e22628[i]*e22628e[i]/e34428[i];
if e22613[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999,-1,-2) and e22628[i]>=0 and e34428[i]>0 then
    othpay1[i]=e22628[i]/e34428[i];
if e22613[i]=4 and e22628[i]>=0 and e34428[i]>0 then othpay1[i]=e22628[i]/(2*e34428[i]);
if e22613[i]=5 and e22628[i]>=0 and e34428[i]>0 then othpay1[i]=e22628[i]/(4.3*e34428[i]);
if e22613[i]=6 and e22628[i]>=0 and e34428[i]>0 and e35600[i]>0 then othpay1[i]=e22628[i]/(e35600[i]*e34428[i]);
if e22613[i]=8 and e22628[i]>=0 and e34428[i]>0 then othpay1[i]=e22628[i]/(2.15*e34428[i]);
if e22613[i] in (12,13) and e22628[i]>=0 then do;
    /* condition based on the time unit to finish per othpay1. */
    if e22628c[i]=3 and e22628cb[i]>0 and e22628e[i]>0 and e34428[i]>0 then
        othpay1[i]=(e22628[i]/e22628cb[i])*e22628e[i]/e34428[i];
    if e22628c[i]=4 and e22628cb[i]>0 and e34428[i]>0 then othpay1[i]=(e22628[i]/e22628cb[i])/e34428[i];
    if e22628c[i]=5 and e22628cb[i]>0 and e34428[i]>0 then othpay1[i]=(e22628[i]/e22628cb[i])/(e34428[i]*4.3);
    /* missing value */
    if e22628c[i] in (3,4,5) and -4<e34428[i]<=0 then othpay1[i]=-3;
end;
if e22613[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34428[i]=0 then othpay1[i]=-3;
if e22613[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34428[i]<0 then
    othpay1[i]=e34428[i];

/* non-hourly wage at the beginning, without overtime*/
if e22613[i]=2 and e22628[i]>=0 and e22628e[i]>0 and e34402[i]>0 then othpay1[i]=e22628[i]*e22628e[i]/e34402[i];
if e22613[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999,-1,-2) and e22628[i]>=0 and e34402[i]>0 then
    othpay1[i]=e22628[i]/e34402[i];
if e22613[i]=4 and e22628[i]>=0 and e34402[i]>0 then othpay1[i]=e22628[i]/(2*e34402[i]);
if e22613[i]=5 and e22628[i]>=0 and e34402[i]>0 then othpay1[i]=e22628[i]/(4.3*e34402[i]);
if e22613[i]=6 and e22628[i]>=0 and e34402[i]>0 and e35600[i]>0 then othpay1[i]=e22628[i]/(e35600[i]*e34402[i]);
if e22613[i]=8 and e22628[i]>=0 and e34402[i]>0 then othpay1[i]=e22628[i]/(2.15*e34402[i]);
if e22613[i] in (12,13) and e22628[i]>=0 then do;
    /* condition based on the time unit to finish per othpay1. */
    if e22628c[i]=3 and e22628cb[i]>0 and e22628e[i]>0 and e34402[i]>0 then
        othpay1[i]=(e22628[i]/e22628cb[i])*e22628e[i]/e34402[i];
    if e22628c[i]=4 and e22628cb[i]>0 and e34402[i]>0 then othpay1[i]=(e22628[i]/e22628cb[i])/e34402[i];
    if e22628c[i]=5 and e22628cb[i]>0 and e34402[i]>0 then othpay1[i]=(e22628[i]/e22628cb[i])/(e34402[i]*4.3);
    /* missing value */
    if e22628c[i] in (3,4,5) and -4<e34402[i]<=0 then othpay1[i]=-3;
end;
if e22613[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay1[i]=-3;
if e22613[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
    othpay1[i]=e34402[i];

if e22613[i]=6 and e35600[i] le 0 then othpay1[i]=-3;
if e22613[i]=6 and -4<e35600[i]<0 then othpay1[i]=e35600[i];

```

```

end;
end;

/***** for commissions *****/
/*report hourly wage at the beginning*/
do i=1 to 9;
if e216002[i]=1 and e225002[i]>=0 then othpay2[i]=e225002[i];
if e23901[i]>0 and e216002[i]=2 and e225502[i]>0 and e225002[i]>=0 then
    othpay2[i]=(e225002[i]*e225502[i])/e23901[i];
if e23901[i]>0 and e216002[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e225002[i]>=0
    then othpay2[i]=e225002[i]/e23901[i];
if e23901[i]>0 and e216002[i]=4 and e225002[i]>=0 then othpay2[i]=e225002[i]/(2*e23901[i]);
if e23901[i]>0 and e216002[i]=5 and e225002[i]>=0 then othpay2[i]=e225002[i]/(4.3*e23901[i]);
if e23901[i]>0 and e216002[i]=6 and e225002[i]>=0 and e35600[i]>0 then
    othpay2[i]=e225002[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e216002[i]=8 and e225002[i]>=0 then othpay2[i]=e225002[i]/(2.15*e23901[i]);
if e216002[i] in (9,14) then othpay2[i]=0;
if e216002[i] in (12,13) and e225002[i]>=0 then do;
    /* condition based on the time unit to finish per othpay2. */
    if e225302[i]=1 and e225352[i]>0 then othpay2[i]=(e225002[i]/e225352[i])*60;
    if e225302[i]=2 and e225352[i]>0 then othpay2[i]=e225002[i]/e225352[i];
    if e225302[i]=3 and e225352[i]>0 and e225502[i]>=0 and e23901[i]>0 then
        othpay2[i]=(e225002[i]/e225352[i])*e225502[i]/e23901[i];
    if e225302[i]=4 and e225352[i]>0 and e23901[i]>0 then othpay2[i]=(e225002[i]/e225352[i])/e23901[i];
    if e225302[i]=5 and e225352[i]>0 and e23901[i]>0 then othpay2[i]=(e225002[i]/e225352[i])/(e23901[i]*4.3);
/* missing value */
    if -4<e225302[i]<0 or -4<e225352[i]<=0 then othpay2[i]=-3;
    if e225302[i] in (3,4,5) and -4<e23901[i]<=0 then othpay2[i]=-3;
    if e225302[i]=3 and e225502[i]<0 then othpay2[i]=-3;
end;
if e216002[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999, -1, -2) and -4<e225002[i]<0 then
    othpay2[i]=e225002[i];
if e216002[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999, -1, -2) and e23901[i]=0 then othpay2[i]=-3;
if e216002[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999, -1, -2) and -4<e23901[i]<0 then
    othpay2[i]=e23901[i];
if e216002[i]=2 and e225502[i] le 0 then othpay2[i]=-3;
if e216002[i]=2 and -4<e225502[i]<0 then othpay2[i]=e225502[i];

end;

/* non-hourly wage at the beginning,with overtime*/
do i=1 to 9;
if e34428[i]>0 and e216002[i]=2 and e225502[i]>0 and e225002[i]>=0 then
    othpay2[i]=(e225002[i]*e225502[i])/e34428[i];
if e34428[i]>0 and e216002[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e225002[i]>=0
    then othpay2[i]=e225002[i]/e34428[i];
if e34428[i]>0 and e216002[i]=4 and e225002[i]>=0 then othpay2[i]=e225002[i]/(2*e34428[i]);
if e34428[i]>0 and e216002[i]=5 and e225002[i]>=0 then othpay2[i]=e225002[i]/(4.3*e34428[i]);
if e34428[i]>0 and e216002[i]=6 and e225002[i]>=0 and e35600[i]>0 then
    othpay2[i]=e225002[i]/(e35600[i]*e34428[i]);
if e34428[i]>0 and e216002[i]=8 and e225002[i]>=0 then othpay2[i]=e225002[i]/(2.15*e34428[i]);
if e216002[i] in (12,13) and e225002[i]>=0 then do;
    /* condition based on the time unit to finish per othpay2. */
    if e225302[i]=3 and e225352[i]>0 and e225502[i]>=0 and e34428[i]>0 then
        othpay2[i]=(e225002[i]/e225352[i])*e225502[i]/e34428[i];

```

```

if e225302[i]=4 and e225352[i]>0 and e34428[i]>0 then othpay2[i]=(e225002[i]/e225352[i])/e34428[i];
if e225302[i]=5 and e225352[i]>0 and e34428[i]>0 then othpay2[i]=(e225002[i]/e225352[i])/(e34428[i]*4.3);
/* missing value */
if e225302[i] in (3,4,5) and -4<e34428[i]<=0 then othpay2[i]=-3;
end;
if e216002[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34428[i]=0 then othpay2[i]=-3;
if e216002[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34428[i]<0 then
    othpay2[i]=e34428[i];

end;

/* non-hourly wage at the beginning, without overtime*/
do i=1 to 9;
if e34402[i]>0 and e216002[i]=2 and e225502[i]>0 and e225002[i]>=0 then
    othpay2[i]=(e225002[i]*e225502[i])/e34402[i];
if e34402[i]>0 and e216002[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e225002[i]>=0
    then othpay2[i]=e225002[i]/e34402[i];
if e34402[i]>0 and e216002[i]=4 and e225002[i]>=0 then othpay2[i]=e225002[i]/(2*e34402[i]);
if e34402[i]>0 and e216002[i]=5 and e225002[i]>=0 then othpay2[i]=e225002[i]/(4.3*e34402[i]);
if e34402[i]>0 and e216002[i]=6 and e225002[i]>=0 and e35600[i]>0 then
    othpay2[i]=e225002[i]/(e35600[i]*e34402[i]);
if e34402[i]>0 and e216002[i]=8 and e225002[i]>=0 then othpay2[i]=e225002[i]/(2.15*e34402[i]);
if e216002[i] in (7,9,12,13,15,16,17,14,18,19,21,22,23,24,25,26,28) then othpf2[i]=othpf2[i]+1;
if e216002[i] in (12,13) and e225002[i]>=0 then do;
    /* condition based on the time unit to finish per othpay2. */
    if e225302[i]=3 and e225352[i]>0 and e225502[i]>=0 and e34402[i]>0 then
        othpay2[i]=(e225002[i]/e225352[i])*e225502[i]/e34402[i];
    if e225302[i]=4 and e225352[i]>0 and e34402[i]>0 then othpay2[i]=(e225002[i]/e225352[i])/e34402[i];
    if e225302[i]=5 and e225352[i]>0 and e34402[i]>0 then othpay2[i]=(e225002[i]/e225352[i])/(e34402[i]*4.3);
    /* missing value */
    if e225302[i] in (3,4,5) and -4<e34402[i]<=0 then othpay2[i]=-3;
end;
if e216002[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay2[i]=-3;
if e216002[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
    othpay2[i]=e34402[i];

if e216002[i]=6 and e35600[i] le 0 then othpay2[i]=-3;
if e216002[i]=6 and -4<e35600[i]<0 then othpay2[i]=e35600[i];
end;

/* if commissions is corrected in the later part */
do i=1 to 9;
if e226046[i]=1 then do;

/*report hourly wage at the beginning*/
if e22614[i]=1 and e22629[i]>=0 then othpay2[i]=e22629[i];
if e22614[i]=2 and e22629[i]>=0 and e22629e[i]>0 and e23901[i]>0 then othpay2[i]=e22629[i]*e22629e[i]/e23901[i];
if e22614[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e22629[i]>=0 and e23901[i]>0 then
    othpay2[i]=e22629[i]/e23901[i];
if e22614[i]=4 and e22629[i]>=0 and e23901[i]>0 then othpay2[i]=e22629[i]/(2*e23901[i]);
if e22614[i]=5 and e22629[i]>=0 and e23901[i]>0 then othpay2[i]=e22629[i]/(4.3*e23901[i]);
if e22614[i]=6 and e22629[i]>=0 and e23901[i]>0 and e35600[i]>0 then othpay2[i]=e22629[i]/(e35600[i]*e23901[i]);
if e22614[i]=8 and e22629[i]>=0 and e23901[i]>0 then othpay2[i]=e22629[i]/(2.15*e23901[i]);
if e22614[i] in (9,14) then othpay2[i]=0;
if e22614[i] in (12,13) and e22629[i]>=0 then do;
    /* condition based on the time unit to finish per othpay2. */
    if e22629c[i]=1 and e22629cb[i]>0 then othpay2[i]=(e22629[i]/e22629cb[i])*60;

```

Appendix 2: Employment Variable Creation

```

if e22629c[i]=2 and e22629cb[i]>0 then othpay2[i]=e22629[i]/e22629cb[i];
if e22629c[i]=3 and e22629cb[i]>0 and e22629e[i]>=0 and e23901[i]>0 then
    othpay2[i]=(e22629[i]/e22629cb[i])*e22629e[i]/e23901[i];
if e22629c[i]=4 and e22629cb[i]>0 and e23901[i]>0 then othpay2[i]=(e22629[i]/e22629cb[i])/e23901[i];
if e22629c[i]=5 and e22629cb[i]>0 and e23901[i]>0 then othpay2[i]=(e22629[i]/e22629cb[i])/(e23901[i]*4.3);
/* missing value */
if -4<e22629c[i]<0 or -4<e22629cb[i]<=0 then othpay2[i]=-3;
if e22629c[i] in (3,4,5) and -4<e23901[i]<=0 then othpay2[i]=-3;
if e22629c[i]=3 and e22629e[i]<0 then othpay2[i]=-3;
end;
if e22614[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e22629[i]<0 then
    othpay2[i]=e22629[i];
if e22614[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay2[i]=-3;
if e22614[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay2[i]=e23901[i];
if e22614[i]=2 and e22629e[i] le 0 then othpay2[i]=-3;
if e22614[i]=2 and -4<e22629e[i]<0 then othpay2[i]=e22629e[i];

/* non-hourly wage at the beginning, with overtime*/
if e22614[i]=2 and e22629[i]>=0 and e22629e[i]>0 and e34428[i]>0 then othpay2[i]=e22629[i]*e22629e[i]/e34428[i];
if e22614[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e22629[i]>=0 and e34428[i]>0 then
    othpay2[i]=e22629[i]/e34428[i];
if e22614[i]=4 and e22629[i]>=0 and e34428[i]>0 then othpay2[i]=e22629[i]/(2*e34428[i]);
if e22614[i]=5 and e22629[i]>=0 and e34428[i]>0 then othpay2[i]=e22629[i]/(4.3*e34428[i]);
if e22614[i]=6 and e22629[i]>=0 and e34428[i]>0 and e35600[i]>0 then othpay2[i]=e22629[i]/(e35600[i]*e34428[i]);
if e22614[i]=8 and e22629[i]>=0 and e34428[i]>0 then othpay2[i]=e22629[i]/(2.15*e34428[i]);
if e22614[i] in (12,13) and e22629[i]>=0 then do;
    /* condition based on the time unit to finish per othpay2. */
    if e22629c[i]=3 and e22629cb[i]>0 and e22629e[i]>=0 and e34428[i]>0 then
        othpay2[i]=(e22629[i]/e22629cb[i])*e22629e[i]/e34428[i];
    if e22629c[i]=4 and e22629cb[i]>0 and e34428[i]>0 then othpay2[i]=(e22629[i]/e22629cb[i])/e34428[i];
    if e22629c[i]=5 and e22629cb[i]>0 and e34428[i]>0 then othpay2[i]=(e22629[i]/e22629cb[i])/(e34428[i]*4.3);
/* missing value */
    if e22629c[i] in (3,4,5) and -4<e34428[i]<=0 then othpay2[i]=-3;
end;
if e22614[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34428[i]=0 then othpay2[i]=-3;
if e22614[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34428[i]<0 then
    othpay2[i]=e34428[i];

/* non-hourly wage at the beginning, without overtime*/
if e22614[i]=2 and e22629[i]>=0 and e22629e[i]>0 and e34402[i]>0 then othpay2[i]=e22629[i]*e22629e[i]/e34402[i];
if e22614[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e22629[i]>=0 and e34402[i]>0 then
    othpay2[i]=e22629[i]/e34402[i];
if e22614[i]=4 and e22629[i]>=0 and e34402[i]>0 then othpay2[i]=e22629[i]/(2*e34402[i]);
if e22614[i]=5 and e22629[i]>=0 and e34402[i]>0 then othpay2[i]=e22629[i]/(4.3*e34402[i]);
if e22614[i]=6 and e22629[i]>=0 and e34402[i]>0 and e35600[i]>0 then othpay2[i]=e22629[i]/(e35600[i]*e34402[i]);
if e22614[i]=8 and e22629[i]>=0 and e34402[i]>0 then othpay2[i]=e22629[i]/(2.15*e34402[i]);
if e22614[i] in (12,13) and e22629[i]>=0 then do;
    /* condition based on the time unit to finish per othpay2. */
    if e22629c[i]=3 and e22629cb[i]>0 and e22629e[i]>=0 and e34402[i]>0 then
        othpay2[i]=(e22629[i]/e22629cb[i])*e22629e[i]/e34402[i];
    if e22629c[i]=4 and e22629cb[i]>0 and e34402[i]>0 then othpay2[i]=(e22629[i]/e22629cb[i])/e34402[i];
    if e22629c[i]=5 and e22629cb[i]>0 and e34402[i]>0 then othpay2[i]=(e22629[i]/e22629cb[i])/(e34402[i]*4.3);
/* missing value */
    if e22629c[i] in (3,4,5) and -4<e34402[i]<=0 then othpay2[i]=-3;
end;
if e22614[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay2[i]=-3;

```

Appendix 2: Employment Variable Creation

```

if e22614[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
    othpay2[i]=e34402[i];

if e22614[i]=6 and e35600[i] le 0 then othpay2[i]=-3;
if e22614[i]=6 and -4<e35600[i]<0 then othpay2[i]=e35600[i];

end;
end;

***** for bonuses *****

/*report hourly wage at the beginning*/
do i=1 to 9;
if e216003[i]=1 and e225003[i]>=0 then othpay3[i]=e225003[i];
if e23901[i]>0 and e216003[i]=2 and e225503[i]>0 and e225003[i]>=0 then
    othpay3[i]=(e225003[i]*e225503[i])/e23901[i];
if e23901[i]>0 and e216003[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e225003[i]>=0
    then othpay3[i]=e225003[i]/e23901[i];
if e23901[i]>0 and e216003[i]=4 and e225003[i]>=0 then othpay3[i]=e225003[i]/(2*e23901[i]);
if e23901[i]>0 and e216003[i]=5 and e225003[i]>=0 then othpay3[i]=e225003[i]/(4.3*e23901[i]);
if e23901[i]>0 and e216003[i]=6 and e225003[i]>=0 and e35600[i]>0 then
    othpay3[i]=e225003[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e216003[i]=8 and e225003[i]>=0 then othpay3[i]=e225003[i]/(2.15*e23901[i]);
if e216003[i] in (9,14) then othpay3[i]=0;
if e216003[i] in (12,13) and e225003[i]>=0 then do;
    /* condition based on the time unit to finish per othpay3. */
    if e225303[i]=1 and e225353[i]>0 then othpay3[i]=(e225003[i]/e225353[i])*60;
    if e225303[i]=2 and e225353[i]>0 then othpay3[i]=e225003[i]/e225353[i];
    if e225303[i]=3 and e225353[i]>0 and e225503[i]>=0 and e23901[i]>0 then
        othpay3[i]=(e225003[i]/e225353[i])*e225503[i]/e23901[i];
    if e225303[i]=4 and e225353[i]>0 and e23901[i]>0 then othpay3[i]=(e225003[i]/e225353[i])/e23901[i];
    if e225303[i]=5 and e225353[i]>0 and e23901[i]>0 then othpay3[i]=(e225003[i]/e225353[i])/(e23901[i]*4.3);
    /* missing value */
    if -4<e225303[i]<0 or -4<e225353[i]<=0 then othpay3[i]=-3;
    if e225303[i] in (3,4,5) and -4<e23901[i]<=0 then othpay3[i]=-3;
    if e225303[i]=3 and e225503[i]<0 then othpay3[i]=-3;
end;
if e216003[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e225003[i]<0 then
    othpay3[i]=e225003[i];
if e216003[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay3[i]=-3;
if e216003[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay3[i]=e23901[i];
if e216003[i]=2 and e225503[i] le 0 then othpay3[i]=-3;
if e216003[i]=2 and -4<e225503[i]<0 then othpay3[i]=e225503[i];
end;

/* non-hourly wage at the beginning, with overtime*/
do i=1 to 9;
if e34428[i]>0 and e216003[i]=2 and e225503[i]>0 and e225003[i]>=0 then
    othpay3[i]=(e225003[i]*e225503[i])/e34428[i];
if e34428[i]>0 and e216003[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e225003[i]>=0
    then othpay3[i]=e225003[i]/e34428[i];
if e34428[i]>0 and e216003[i]=4 and e225003[i]>=0 then othpay3[i]=e225003[i]/(2*e34428[i]);
if e34428[i]>0 and e216003[i]=5 and e225003[i]>=0 then othpay3[i]=e225003[i]/(4.3*e34428[i]);
if e34428[i]>0 and e216003[i]=6 and e225003[i]>=0 and e35600[i]>0 then
    othpay3[i]=e225003[i]/(e35600[i]*e34428[i]);
if e34428[i]>0 and e216003[i]=8 and e225003[i]>=0 then othpay3[i]=e225003[i]/(2.15*e34428[i]);

```

```

if e216003[i] in (12,13) and e225003[i]>=0 then do;
/* condition based on the time unit to finish per othpay3. */
  if e225303[i]=3 and e225353[i]>0 and e225503[i]>=0 and e34428[i]>0 then
    othpay3[i]=(e225003[i]/e225353[i])*e225503[i]/e34428[i];
  if e225303[i]=4 and e225353[i]>0 and e34428[i]>0 then othpay3[i]=(e225003[i]/e225353[i])/e34428[i];
  if e225303[i]=5 and e225353[i]>0 and e34428[i]>0 then othpay3[i]=(e225003[i]/e225353[i])/(e34428[i]*4.3);
/* missing value */
  if e225303[i] in (3,4,5) and -4<e34428[i]<=0 then othpay3[i]=-3;
end;
if e216003[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34428[i]=0 then othpay3[i]=-3;
if e216003[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34428[i]<0 then
  othpay3[i]=e34428[i];
end;

/* non-hourly wage at the beginning,without overtime*/
do i=1 to 9;
if e34402[i]>0 and e216003[i]=2 and e225503[i]>0 and e225003[i]>=0 then
  othpay3[i]=(e225003[i]*e225503[i])/e34402[i];
if e34402[i]>0 and e216003[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e225003[i]>=0
  then othpay3[i]=e225003[i]/e34402[i];
if e34402[i]>0 and e216003[i]=4 and e225003[i]>=0 then othpay3[i]=e225003[i]/(2*e34402[i]);
if e34402[i]>0 and e216003[i]=5 and e225003[i]>=0 then othpay3[i]=e225003[i]/(4.3*e34402[i]);
if e34402[i]>0 and e216003[i]=6 and e225003[i]>=0 and e35600[i]>0 then
  othpay3[i]=e225003[i]/(e35600[i]*e34402[i]);
if e34402[i]>0 and e216003[i]=8 and e225003[i]>=0 then othpay3[i]=e225003[i]/(2.15*e34402[i]);
if e216003[i] in (7,9,12,13,15,16,17,14,18,19,21,22,23,24,25,26,28) then othpf3[i]=othpf3[i]+1;
if e216003[i] in (12,13) and e225003[i]>=0 then do;
  /* condition based on the time unit to finish per othpay3. */
  if e225303[i]=3 and e225353[i]>0 and e225503[i]>=0 and e34402[i]>0 then
    othpay3[i]=(e225003[i]/e225353[i])*e225503[i]/e34402[i];
  if e225303[i]=4 and e225353[i]>0 and e34402[i]>0 then othpay3[i]=(e225003[i]/e225353[i])/e34402[i];
  if e225303[i]=5 and e225353[i]>0 and e34402[i]>0 then othpay3[i]=(e225003[i]/e225353[i])/(e34402[i]*4.3);
/* missing value */
  if e225303[i] in (3,4,5) and -4<e34402[i]<=0 then othpay3[i]=-3;
end;
if e216003[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay3[i]=-3;
if e216003[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
  othpay3[i]=e34402[i];
if e216003[i]=6 and e35600[i] le 0 then othpay3[i]=-3;
if e216003[i]=6 and -4<e35600[i]<0 then othpay3[i]=e35600[i];
end;

/* if bonus is corrected in the later part */
do i=1 to 9;
if e226047[i]=1 then do;

/*report hourly wage at the beginning*/
if e22615[i]=1 and e22630[i]>=0 then othpay3[i]=e22630[i];
if e22615[i]=2 and e22630[i]>=0 and e22630e[i]>0 and e23901[i]>0 then othpay3[i]=e22630[i]*e22630e[i]/e23901[i];
if e22615[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e22630[i]>=0 and e23901[i]>0 then
  othpay3[i]=e22630[i]/e23901[i];
if e22615[i]=4 and e22630[i]>=0 and e23901[i]>0 then othpay3[i]=e22630[i]/(2*e23901[i]);
if e22615[i]=5 and e22630[i]>=0 and e23901[i]>0 then othpay3[i]=e22630[i]/(4.3*e23901[i]);
if e22615[i]=6 and e22630[i]>=0 and e23901[i]>0 and e35600[i]>0 then othpay3[i]=e22630[i]/(e35600[i]*e23901[i]);
if e22615[i]=8 and e22630[i]>=0 and e23901[i]>0 then othpay3[i]=e22630[i]/(2.15*e23901[i]);
if e22615[i] in (9,14) then othpay3[i]=0;
if e22615[i] in (12,13) and e22630[i]>=0 then do;

```

```

/* condition based on the time unit to finish per othpay3. */
if e22630c[i]=1 and e22630cb[i]>0 then othpay3[i]=(e22630[i]/e22630cb[i])*60;
if e22630c[i]=2 and e22630cb[i]>0 then othpay3[i]=e22630[i]/e22630cb[i];
if e22630c[i]=3 and e22630cb[i]>0 and e22630e[i]>=0 and e23901[i]>0 then
    othpay3[i]=(e22630[i]/e22630cb[i])*e22630e[i]/e23901[i];
if e22630c[i]=4 and e22630cb[i]>0 and e23901[i]>0 then othpay3[i]=(e22630[i]/e22630cb[i])/e23901[i];
if e22630c[i]=5 and e22630cb[i]>0 and e23901[i]>0 then othpay3[i]=(e22630[i]/e22630cb[i])/(e23901[i]*4.3);
/* missing value */
if -4<e22630c[i]<0 or -4<e22630cb[i]<=0 then othpay3[i]=-3;
if e22630c[i] in (3,4,5) and -4<e23901[i]<=0 then othpay3[i]=-3;
if e22630c[i]=3 and e22630e[i]<0 then othpay3[i]=-3;
end;
if e22615[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e22630[i]<0 then
    othpay3[i]=e22630[i];
if e22615[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay3[i]=-3;
if e22615[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay3[i]=e23901[i];
if e22615[i]=2 and e22630e[i] le 0 then othpay3[i]=-3;
if e22615[i]=2 and -4<e22630e[i]<0 then othpay3[i]=e22630e[i];

/* non-hourly wage at the beginning, with overtime*/
if e22615[i]=2 and e22630[i]>=0 and e22630e[i]>0 and e34428[i]>0 then othpay3[i]=e22630[i]*e22630e[i]/e34428[i];
if e22615[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999,-1,-2) and e22630[i]>=0 and e34428[i]>0 then
    othpay3[i]=e22630[i]/e34428[i];
if e22615[i]=4 and e22630[i]>=0 and e34428[i]>0 then othpay3[i]=e22630[i]/(2*e34428[i]);
if e22615[i]=5 and e22630[i]>=0 and e34428[i]>0 then othpay3[i]=e22630[i]/(4.3*e34428[i]);
if e22615[i]=6 and e22630[i]>=0 and e34428[i]>0 and e35600[i]>0 then othpay3[i]=e22630[i]/(e35600[i]*e34428[i]);
if e22615[i]=8 and e22630[i]>=0 and e34428[i]>0 then othpay3[i]=e22630[i]/(2.15*e34428[i]);
if e22615[i] in (12,13) and e22630[i]>=0 then do;
    /* condition based on the time unit to finish per othpay3. */
    if e22630c[i]=3 and e22630cb[i]>0 and e22630e[i]>=0 and e34428[i]>0 then
        othpay3[i]=(e22630[i]/e22630cb[i])*e22630e[i]/e34428[i];
    if e22630c[i]=4 and e22630cb[i]>0 and e34428[i]>0 then othpay3[i]=(e22630[i]/e22630cb[i])/e34428[i];
    if e22630c[i]=5 and e22630cb[i]>0 and e34428[i]>0 then othpay3[i]=(e22630[i]/e22630cb[i])/(e34428[i]*4.3);
    /* missing value */
    if e22630c[i] in (3,4,5) and -4<e34428[i]<=0 then othpay3[i]=-3;
end;
if e22615[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34428[i]=0 then othpay3[i]=-3;
if e22615[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34428[i]<0 then
    othpay3[i]=e34428[i];

/* non-hourly wage at the beginning, without overtime*/
if e22615[i]=2 and e22630[i]>=0 and e22630e[i]>0 and e34402[i]>0 then othpay3[i]=e22630[i]*e22630e[i]/e34402[i];
if e22615[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999,-1,-2) and e22630[i]>=0 and e34402[i]>0 then
    othpay3[i]=e22630[i]/e34402[i];
if e22615[i]=4 and e22630[i]>=0 and e34402[i]>0 then othpay3[i]=e22630[i]/(2*e34402[i]);
if e22615[i]=5 and e22630[i]>=0 and e34402[i]>0 then othpay3[i]=e22630[i]/(4.3*e34402[i]);
if e22615[i]=6 and e22630[i]>=0 and e34402[i]>0 and e35600[i]>0 then othpay3[i]=e22630[i]/(e35600[i]*e34402[i]);
if e22615[i]=8 and e22630[i]>=0 and e34402[i]>0 then othpay3[i]=e22630[i]/(2.15*e34402[i]);
if e22615[i] in (12,13) and e22630[i]>=0 then do;
    /* condition based on the time unit to finish per othpay3. */
    if e22630c[i]=3 and e22630cb[i]>0 and e22630e[i]>=0 and e34402[i]>0 then
        othpay3[i]=(e22630[i]/e22630cb[i])*e22630e[i]/e34402[i];
    if e22630c[i]=4 and e22630cb[i]>0 and e34402[i]>0 then othpay3[i]=(e22630[i]/e22630cb[i])/e34402[i];
    if e22630c[i]=5 and e22630cb[i]>0 and e34402[i]>0 then othpay3[i]=(e22630[i]/e22630cb[i])/(e34402[i]*4.3);
    /* missing value */
    if e22630c[i] in (3,4,5) and -4<e34402[i]<=0 then othpay3[i]=-3;

```

Appendix 2: Employment Variable Creation

```

end;
if e22615[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay3[i]=-3;
if e22615[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
    othpay3[i]=e34402[i];

if e22615[i]=6 and e35600[i] le 0 then othpay3[i]=-3;
if e22615[i]=6 and -4<e35600[i]<0 then othpay3[i]=e35600[i];
end;
end;

/****** for incentive pay *****/

/* report hourly wage at the beginning*/
do i=1 to 9;
if e216004[i]=1 and e225004[i]>=0 then othpay4[i]=e225004[i];
if e23901[i]>0 and e216004[i]=2 and e225504[i]>0 and e225004[i]>=0 then
    othpay4[i]=(e225004[i]*e225504[i])/e23901[i];
if e23901[i]>0 and e216004[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e225004[i]>=0
    then othpay4[i]=e225004[i]/e23901[i];
if e23901[i]>0 and e216004[i]=4 and e225004[i]>=0 then othpay4[i]=e225004[i]/(2*e23901[i]);
if e23901[i]>0 and e216004[i]=5 and e225004[i]>=0 then othpay4[i]=e225004[i]/(4.3*e23901[i]);
if e23901[i]>0 and e216004[i]=6 and e225004[i]>=0 and e35600[i]>0 then
    othpay4[i]=e225004[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e216004[i]=8 and e225004[i]>=0 then othpay4[i]=e225004[i]/(2.15*e23901[i]);
if e216004[i] in (9,14) then othpay4[i]=0;
if e216004[i] in (12,13) and e225004[i]>=0 then do;
    /* condition based on the time unit to finish per othpay4. */
    if e225304[i]=1 and e225354[i]>0 then othpay4[i]=(e225004[i]/e225354[i])*60;
    if e225304[i]=2 and e225354[i]>0 then othpay4[i]=e225004[i]/e225354[i];
    if e225304[i]=3 and e225354[i]>0 and e225504[i]>=0 and e23901[i]>0 then
        othpay4[i]=(e225004[i]/e225354[i])*e225504[i]/e23901[i];
    if e225304[i]=4 and e225354[i]>0 and e23901[i]>0 then othpay4[i]=(e225004[i]/e225354[i])/e23901[i];
    if e225304[i]=5 and e225354[i]>0 and e23901[i]>0 then othpay4[i]=(e225004[i]/e225354[i])/(e23901[i]*4.3);
/* missing value */
    if -4<e225304[i]<0 or -4<e225354[i]<=0 then othpay4[i]=-3;
    if e225304[i] in (3,4,5) and -4<e23901[i]<=0 then othpay4[i]=-3;
    if e225304[i]=3 and e225504[i]<0 then othpay4[i]=-3;
end;
if e216004[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e225004[i]<0 then
    othpay4[i]=e225004[i];
if e216004[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay4[i]=-3;
if e216004[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay4[i]=e23901[i];
if e216004[i]=2 and e225504[i] le 0 then othpay4[i]=-3;
if e216004[i]=2 and -4<e225504[i]<0 then othpay4[i]=e225504[i];

end;

/* non-hourly wage at the beginning, with overtime*/
do i=1 to 9;
if e34428[i]>0 and e216004[i]=2 and e225504[i]>0 and e225004[i]>=0 then
    othpay4[i]=(e225004[i]*e225504[i])/e34428[i];
if e34428[i]>0 and e216004[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e225004[i]>=0
    then othpay4[i]=e225004[i]/e34428[i];
if e34428[i]>0 and e216004[i]=4 and e225004[i]>=0 then othpay4[i]=e225004[i]/(2*e34428[i]);
if e34428[i]>0 and e216004[i]=5 and e225004[i]>=0 then othpay4[i]=e225004[i]/(4.3*e34428[i]);

```

```

if e34428[i]>0 and e216004[i]=6 and e225004[i]>=0 and e35600[i]>0 then
    othpay4[i]=e225004[i]/(e35600[i]*e34428[i]);
if e34428[i]>0 and e216004[i]=8 and e225004[i]>=0 then othpay4[i]=e225004[i]/(2.15*e34428[i]);
if e216004[i] in (12,13) and e225004[i]>=0 then do;
    /* condition based on the time unit to finish per othpay4. */
    if e225304[i]=3 and e225354[i]>0 and e225504[i]>=0 and e34428[i]>0 then
        othpay4[i]=(e225004[i]/e225354[i])*e225504[i]/e34428[i];
    if e225304[i]=4 and e225354[i]>0 and e34428[i]>0 then othpay4[i]=(e225004[i]/e225354[i])/e34428[i];
    if e225304[i]=5 and e225354[i]>0 and e34428[i]>0 then othpay4[i]=(e225004[i]/e225354[i])/(e34428[i]*4.3);
    /* missing value */
    if e225304[i] in (3,4,5) and -4<e34428[i]<=0 then othpay4[i]=-3;
end;
if e216004[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34428[i]=0 then othpay4[i]=-3;
if e216004[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34428[i]<0 then
    othpay4[i]=e34428[i];
end;

/* non-hourly wage at the beginning, without overtime*/
do i=1 to 9;
if e34402[i]>0 and e216004[i]=2 and e225504[i]>0 and e225004[i]>=0 then
    othpay4[i]=(e225004[i]*e225504[i])/e34402[i];
if e34402[i]>0 and e216004[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e225004[i]>=0
    then othpay4[i]=e225004[i]/e34402[i];
if e34402[i]>0 and e216004[i]=4 and e225004[i]>=0 then othpay4[i]=e225004[i]/(2*e34402[i]);
if e34402[i]>0 and e216004[i]=5 and e225004[i]>=0 then othpay4[i]=e225004[i]/(4.3*e34402[i]);
if e34402[i]>0 and e216004[i]=6 and e225004[i]>=0 and e35600[i]>0 then
    othpay4[i]=e225004[i]/(e35600[i]*e34402[i]);
if e34402[i]>0 and e216004[i]=8 and e225004[i]>=0 then othpay4[i]=e225004[i]/(2.15*e34402[i]);
if e216004[i] in (7,9,12,13,15,16,17,14,18,19,21,22,23,24,25,26,28) then othpf4[i]=othpf4[i]+1;
if e216004[i] in (12,13) and e225004[i]>=0 then do;
    /* condition based on the time unit to finish per othpay4. */
    if e225304[i]=3 and e225354[i]>0 and e225504[i]>=0 and e34402[i]>0 then
        othpay4[i]=(e225004[i]/e225354[i])*e225504[i]/e34402[i];
    if e225304[i]=4 and e225354[i]>0 and e34402[i]>0 then othpay4[i]=(e225004[i]/e225354[i])/e34402[i];
    if e225304[i]=5 and e225354[i]>0 and e34402[i]>0 then othpay4[i]=(e225004[i]/e225354[i])/(e34402[i]*4.3);
    /* missing value */
    if e225304[i] in (3,4,5) and -4<e34402[i]<=0 then othpay4[i]=-3;
end;
if e216004[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay4[i]=-3;
if e216004[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
    othpay4[i]=e34402[i];

if e216004[i]=6 and e35600[i] le 0 then othpay4[i]=-3;
if e216004[i]=6 and -4<e35600[i]<0 then othpay4[i]=e35600[i];
end;

/* if incentive pay is corrected in the later part */
do i=1 to 9;
if e226048[i]=1 then do;

/*report hourly wage at the beginning*/
if e22616[i]=1 and e22631[i]>=0 then othpay4[i]=e22631[i];
if e22616[i]=2 and e22631[i]>=0 and e22631e[i]>0 and e23901[i]>0 then othpay4[i]=e22631[i]*e22631e[i]/e23901[i];
if e22616[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e22631[i]>=0 and e23901[i]>0 then
    othpay4[i]=e22631[i]/e23901[i];
if e22616[i]=4 and e22631[i]>=0 and e23901[i]>0 then othpay4[i]=e22631[i]/(2*e23901[i]);

```

Appendix 2: Employment Variable Creation

```

if e22616[i]=5 and e22631[i]>=0 and e23901[i]>0 then othpay4[i]=e22631[i]/(4.3*e23901[i]);
if e22616[i]=6 and e22631[i]>=0 and e23901[i]>0 and e35600[i]>0 then othpay4[i]=e22631[i]/(e35600[i]*e23901[i]);
if e22616[i]=8 and e22631[i]>=0 and e23901[i]>0 then othpay4[i]=e22631[i]/(2.15*e23901[i]);
if e22616[i] in (9,14) then othpay4[i]=0;
if e22616[i] in (12,13) and e22631[i]>=0 then do;
    /* condition based on the time unit to finish per othpay4. */
    if e22631c[i]=1 and e22631cb[i]>0 then othpay4[i]=(e22631[i]/e22631cb[i])*60;
    if e22631c[i]=2 and e22631cb[i]>0 then othpay4[i]=e22631[i]/e22631cb[i];
    if e22631c[i]=3 and e22631cb[i]>0 and e22631e[i]>=0 and e23901[i]>0 then
        othpay4[i]=(e22631[i]/e22631cb[i])*e22631e[i]/e23901[i];
    if e22631c[i]=4 and e22631cb[i]>0 and e23901[i]>0 then othpay4[i]=(e22631[i]/e22631cb[i])/e23901[i];
    if e22631c[i]=5 and e22631cb[i]>0 and e23901[i]>0 then othpay4[i]=(e22631[i]/e22631cb[i])/(e23901[i]*4.3);
    /* missing value */
    if -4<e22631c[i]<0 or -4<e22631cb[i]<=0 then othpay4[i]=-3;
    if e22631c[i] in (3,4,5) and -4<e23901[i]<=0 then othpay4[i]=-3;
    if e22631c[i]=3 and e22631e[i]<0 then othpay4[i]=-3;
end;
if e22616[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e22631[i]<0 then
    othpay4[i]=e22631[i];
if e22616[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay4[i]=-3;
if e22616[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay4[i]=e23901[i];
if e22616[i]=2 and e22631e[i] le 0 then othpay4[i]=-3;
if e22616[i]=2 and -4<e22631e[i]<0 then othpay4[i]=e22631e[i];

/* non-hourly wage at the beginning, with overtime*/
if e22616[i]=2 and e22631[i]>=0 and e22631e[i]>0 and e34428[i]>0 then othpay4[i]=e22631[i]*e22631e[i]/e34428[i];
if e22616[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e22631[i]>=0 and e34428[i]>0 then
    othpay4[i]=e22631[i]/e34428[i];
if e22616[i]=4 and e22631[i]>=0 and e34428[i]>0 then othpay4[i]=e22631[i]/(2*e34428[i]);
if e22616[i]=5 and e22631[i]>=0 and e34428[i]>0 then othpay4[i]=e22631[i]/(4.3*e34428[i]);
if e22616[i]=6 and e22631[i]>=0 and e34428[i]>0 and e35600[i]>0 then othpay4[i]=e22631[i]/(e35600[i]*e34428[i]);
if e22616[i]=8 and e22631[i]>=0 and e34428[i]>0 then othpay4[i]=e22631[i]/(2.15*e34428[i]);
if e22616[i] in (12,13) and e22631[i]>=0 then do;
    /* condition based on the time unit to finish per othpay4. */
    if e22631c[i]=3 and e22631cb[i]>0 and e22631e[i]>=0 and e34428[i]>0 then
        othpay4[i]=(e22631[i]/e22631cb[i])*e22631e[i]/e34428[i];
    if e22631c[i]=4 and e22631cb[i]>0 and e34428[i]>0 then othpay4[i]=(e22631[i]/e22631cb[i])/e34428[i];
    if e22631c[i]=5 and e22631cb[i]>0 and e34428[i]>0 then othpay4[i]=(e22631[i]/e22631cb[i])/(e34428[i]*4.3);
    /* missing value */
    if e22631c[i] in (3,4,5) and -4<e34428[i]<=0 then othpay4[i]=-3;
end;
if e22616[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34428[i]=0 then othpay4[i]=-3;
if e22616[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34428[i]<0 then
    othpay4[i]=e34428[i];

/* non-hourly wage at the beginning, without overtime*/
if e22616[i]=2 and e22631[i]>=0 and e22631e[i]>0 and e34402[i]>0 then othpay4[i]=e22631[i]*e22631e[i]/e34402[i];
if e22616[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e22631[i]>=0 and e34402[i]>0 then
    othpay4[i]=e22631[i]/e34402[i];
if e22616[i]=4 and e22631[i]>=0 and e34402[i]>0 then othpay4[i]=e22631[i]/(2*e34402[i]);
if e22616[i]=5 and e22631[i]>=0 and e34402[i]>0 then othpay4[i]=e22631[i]/(4.3*e34402[i]);
if e22616[i]=6 and e22631[i]>=0 and e34402[i]>0 and e35600[i]>0 then othpay4[i]=e22631[i]/(e35600[i]*e34402[i]);
if e22616[i]=8 and e22631[i]>=0 and e34402[i]>0 then othpay4[i]=e22631[i]/(2.15*e34402[i]);
if e22616[i] in (12,13) and e22631[i]>=0 then do;
    /* condition based on the time unit to finish per othpay4. */
    if e22631c[i]=3 and e22631cb[i]>0 and e22631e[i]>=0 and e34402[i]>0 then

```

```

othpay4[i]=(e22631[i]/e22631cb[i])*e22631e[i]/e34402[i];
if e22631c[i]=4 and e22631cb[i]>0 and e34402[i]>0 then othpay4[i]=(e22631[i]/e22631cb[i])/e34402[i];
if e22631c[i]=5 and e22631cb[i]>0 and e34402[i]>0 then othpay4[i]=(e22631[i]/e22631cb[i])/(e34402[i]*4.3);
/* missing value */
if e22631c[i] in (3,4,5) and -4<e34402[i]<=0 then othpay4[i]=-3;
end;
if e22616[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay4[i]=-3;
if e22616[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
    othpay4[i]=e34402[i];

if e22616[i]=6 and e35600[i] le 0 then othpay4[i]=-3;
if e22616[i]=6 and -4<e35600[i]<0 then othpay4[i]=e35600[i];
end;
end;

***** for others *****

/*report hourly wage at the beginning*/
do i=1 to 9;
if e216005[i]=1 and e225005[i]>0 then othpay5[i]=e225005[i];
if e23901[i]>0 and e216005[i]=2 and e225505[i]>0 and e225005[i]>=0 then
    othpay5[i]=(e225005[i]*e225505[i])/e23901[i];
if e23901[i]>0 and e216005[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e225005[i]>=0
    then othpay5[i]=e225005[i]/e23901[i];
if e23901[i]>0 and e216005[i]=4 and e225005[i]>=0 then othpay5[i]=e225005[i]/(2*e23901[i]);
if e23901[i]>0 and e216005[i]=5 and e225005[i]>=0 then othpay5[i]=e225005[i]/(4.3*e23901[i]);
if e23901[i]>0 and e216005[i]=6 and e225005[i]>=0 and e35600[i]>0 then
    othpay5[i]=e225005[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e216005[i]=8 and e225005[i]>=0 then othpay5[i]=e225005[i]/(2.15*e23901[i]);
if e216005[i] in (9,14) then othpay5[i]=0;
if e216005[i] in (12,13) and e225005[i]>=0 then do;
    /* condition based on the time unit to finish per othpay5. */
    if e225305[i]=1 and e225355[i]>0 then othpay5[i]=(e225005[i]/e225355[i])*60;
    if e225305[i]=2 and e225355[i]>0 then othpay5[i]=e225005[i]/e225355[i];
    if e225305[i]=3 and e225355[i]>0 and e225505[i]>=0 and e23901[i]>0 then
        othpay5[i]=(e225005[i]/e225355[i])*e225505[i]/e23901[i];
    if e225305[i]=4 and e225355[i]>0 and e23901[i]>0 then othpay5[i]=(e225005[i]/e225355[i])/e23901[i];
    if e225305[i]=5 and e225355[i]>0 and e23901[i]>0 then othpay5[i]=(e225005[i]/e225355[i])/(e23901[i]*4.3);
    /* missing value */
    if -4<e225305[i]<0 or -4<e225355[i]<=0 then othpay5[i]=-3;
    if e225305[i] in (3,4,5) and -4<e23901[i]<=0 then othpay5[i]=-3;
    if e225305[i]=3 and e225505[i]<0 then othpay5[i]=-3;
end;
if e216005[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e225005[i]<0 then
    othpay5[i]=e225005[i];
if e216005[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay5[i]=-3;
if e216005[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay5[i]=e23901[i];
if e216005[i]=2 and e225505[i] le 0 then othpay5[i]=-3;
if e216005[i]=2 and -4<e225505[i]<0 then othpay5[i]=e225505[i];
end;

/* non-hourly wage at the beginning, with overtime*/
do i=1 to 9;
if e34428[i]>0 and e216005[i]=2 and e225505[i]>0 and e225005[i]>=0 then
    othpay5[i]=(e225005[i]*e225505[i])/e34428[i];

```

```

if e34428[i]>0 and e216005[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e225005[i]>=0
    then othpay5[i]=e225005[i]/e34428[i];
if e34428[i]>0 and e216005[i]=4 and e225005[i]>=0 then othpay5[i]=e225005[i]/(2*e34428[i]);
if e34428[i]>0 and e216005[i]=5 and e225005[i]>=0 then othpay5[i]=e225005[i]/(4.3*e34428[i]);
if e34428[i]>0 and e216005[i]=6 and e225005[i]>=0 and e35600[i]>0 then
    othpay5[i]=e225005[i]/(e35600[i]*e34428[i]);
if e34428[i]>0 and e216005[i]=8 and e225005[i]>=0 then othpay5[i]=e225005[i]/(2.15*e34428[i]);
if e216005[i] in (12, 13) and e225005[i]>=0 then do;
    /* condition based on the time unit to finish per othpay5. */
    if e225305[i]=3 and e225355[i]>0 and e225505[i]>=0 and e34428[i]>0 then
        othpay5[i]=(e225005[i]/e225355[i])*e225505[i]/e34428[i];
    if e225305[i]=4 and e225355[i]>0 and e34428[i]>0 then othpay5[i]=(e225005[i]/e225355[i])/e34428[i];
    if e225305[i]=5 and e225355[i]>0 and e34428[i]>0 then othpay5[i]=(e225005[i]/e225355[i])/(e34428[i]*4.3);
    /* missing value */
    if e225305[i] in (3, 4, 5) and -4<e34428[i]<=0 then othpay5[i]=-3;
end;
if e216005[i] in (2, 3, 4, 5, 6, 7, 8, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e34428[i]=0 then othpay5[i]=-3;
if e216005[i] in (2, 3, 4, 5, 6, 7, 8, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and -4<e34428[i]<0 then
    othpay5[i]=e34428[i];

end;

/* non-hourly wage at the beginning, without overtime*/
do i=1 to 9;
if e34402[i]>0 and e216005[i]=2 and e225505[i]>0 and e225005[i]>=0 then
    othpay5[i]=(e225005[i]*e225505[i])/e34402[i];
if e34402[i]>0 and e216005[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e225005[i]>=0
    then othpay5[i]=e225005[i]/e34402[i];
if e34402[i]>0 and e216005[i]=4 and e225005[i]>=0 then othpay5[i]=e225005[i]/(2*e34402[i]);
if e34402[i]>0 and e216005[i]=5 and e225005[i]>=0 then othpay5[i]=e225005[i]/(4.3*e34402[i]);
if e34402[i]>0 and e216005[i]=6 and e225005[i]>=0 and e35600[i]>0 then
    othpay5[i]=e225005[i]/(e35600[i]*e34402[i]);
if e34402[i]>0 and e216005[i]=8 and e225005[i]>=0 then othpay5[i]=e225005[i]/(2.15*e34402[i]);
if e216005[i] in (7, 9, 12, 13, 15, 16, 17, 14, 18, 19, 21, 22, 23, 24, 25, 26, 28) then othpf5[i]=othpf5[i]+1;
if e216005[i] in (12, 13) and e225005[i]>=0 then do;
    /* condition based on the time unit to finish per othpay5. */
    if e225305[i]=3 and e225355[i]>0 and e225505[i]>=0 and e34402[i]>0 then
        othpay5[i]=(e225005[i]/e225355[i])*e225505[i]/e34402[i];
    if e225305[i]=4 and e225355[i]>0 and e34402[i]>0 then othpay5[i]=(e225005[i]/e225355[i])/e34402[i];
    if e225305[i]=5 and e225355[i]>0 and e34402[i]>0 then othpay5[i]=(e225005[i]/e225355[i])/(e34402[i]*4.3);
    /* missing value */
    if e225305[i] in (3, 4, 5) and -4<e34402[i]<=0 then othpay5[i]=-3;
end;
if e216005[i] in (2, 3, 4, 5, 6, 7, 8, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e34402[i]=0 then othpay5[i]=-3;
if e216005[i] in (2, 3, 4, 5, 6, 7, 8, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and -4<e34402[i]<0 then
    othpay5[i]=e34402[i];

if e216005[i]=6 and e35600[i] le 0 then othpay5[i]=-3;
if e216005[i]=6 and -4<e35600[i]<0 then othpay5[i]=e35600[i];

end;

/* if other compensation is corrected in the later part */
do i=1 to 9;
if e226049[i]=1 then do;

/*report hourly wage at the beginnnig*/

```

```

if e22617[i]=1 and e22632[i]>=0 then othpay5[i]=e22632[i];
if e22617[i]=2 and e22632[i]>=0 and e22632e[i]>0 and e23901[i]>0 then othpay5[i]=e22632[i]*e22632e[i]/e23901[i];
if e22617[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e22632[i]>=0 and e23901[i]>0 then
    othpay5[i]=e22632[i]/e23901[i];
if e22617[i]=4 and e22632[i]>=0 and e23901[i]>0 then othpay5[i]=e22632[i]/(2*e23901[i]);
if e22617[i]=5 and e22632[i]>=0 and e23901[i]>0 then othpay5[i]=e22632[i]/(4.3*e23901[i]);
if e22617[i]=6 and e22632[i]>=0 and e23901[i]>0 and e35600[i]>0 then othpay5[i]=e22632[i]/(e35600[i]*e23901[i]);
if e22617[i]=8 and e22632[i]>=0 and e23901[i]>0 then othpay5[i]=e22632[i]/(2.15*e23901[i]);
if e22617[i] in (9,14) then othpay5[i]=0;
if e22617[i] in (12,13) and e22632[i]>=0 then do;
    /* condition based on the time unit to finish per othpay5. */
    if e22632c[i]=1 and e22632cb[i]>0 then othpay5[i]=(e22632[i]/e22632cb[i])*60;
    if e22632c[i]=2 and e22632cb[i]>0 then othpay5[i]=e22632[i]/e22632cb[i];
    if e22632c[i]=3 and e22632cb[i]>0 and e22632e[i]>=0 and e23901[i]>0 then
        othpay5[i]=(e22632[i]/e22632cb[i])*e22632e[i]/e23901[i];
    if e22632c[i]=4 and e22632cb[i]>0 and e23901[i]>0 then othpay5[i]=(e22632[i]/e22632cb[i])/e23901[i];
    if e22632c[i]=5 and e22632cb[i]>0 and e23901[i]>0 then othpay5[i]=(e22632[i]/e22632cb[i])/(e23901[i]*4.3);
    /* missing value */
    if -4<e22632c[i]<0 or -4<e22632cb[i]<=0 then othpay5[i]=-3;
    if e22632c[i] in (3,4,5) and -4<e23901[i]<=0 then othpay5[i]=-3;
    if e22632c[i]=3 and e22632e[i]<0 then othpay5[i]=-3;
end;
if e22617[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e22632[i]<0 then
    othpay5[i]=e22632[i];
if e22617[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay5[i]=-3;
if e22617[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay5[i]=e23901[i];
if e22617[i]=2 and e22632e[i] le 0 then othpay5[i]=-3;
if e22617[i]=2 and -4<e22632e[i]<0 then othpay5[i]=e22632e[i];

/* non-hourly wage at the beginning, with overtime*/
if e22617[i]=2 and e22632[i]>=0 and e22632e[i]>0 and e34428[i]>0 then othpay5[i]=e22632[i]*e22632e[i]/e34428[i];
if e22617[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e22632[i]>=0 and e34428[i]>0 then
    othpay5[i]=e22632[i]/e34428[i];
if e22617[i]=4 and e22632[i]>=0 and e34428[i]>0 then othpay5[i]=e22632[i]/(2*e34428[i]);
if e22617[i]=5 and e22632[i]>=0 and e34428[i]>0 then othpay5[i]=e22632[i]/(4.3*e34428[i]);
if e22617[i]=6 and e22632[i]>=0 and e34428[i]>0 and e35600[i]>0 then othpay5[i]=e22632[i]/(e35600[i]*e34428[i]);
if e22617[i]=8 and e22632[i]>=0 and e34428[i]>0 then othpay5[i]=e22632[i]/(2.15*e34428[i]);
if e22617[i] in (12,13) and e22632[i]>=0 then do;
    /* condition based on the time unit to finish per othpay5. */
    if e22632c[i]=3 and e22632cb[i]>0 and e22632e[i]>=0 and e34428[i]>0 then
        othpay5[i]=(e22632[i]/e22632cb[i])*e22632e[i]/e34428[i];
    if e22632c[i]=4 and e22632cb[i]>0 and e34428[i]>0 then othpay5[i]=(e22632[i]/e22632cb[i])/e34428[i];
    if e22632c[i]=5 and e22632cb[i]>0 and e34428[i]>0 then othpay5[i]=(e22632[i]/e22632cb[i])/(e34428[i]*4.3);
    /* missing value */
    if e22632c[i] in (3,4,5) and -4<e34428[i]<=0 then othpay5[i]=-3;
end;
if e22617[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34428[i]=0 then othpay5[i]=-3;
if e22617[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34428[i]<0 then
    othpay5[i]=e34428[i];

/* non-hourly wage at the beginning, without overtime*/
if e22617[i]=2 and e22632[i]>=0 and e22632e[i]>0 and e34402[i]>0 then othpay5[i]=e22632[i]*e22632e[i]/e34402[i];
if e22617[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e22632[i]>=0 and e34402[i]>0 then
    othpay5[i]=e22632[i]/e34402[i];
if e22617[i]=4 and e22632[i]>=0 and e34402[i]>0 then othpay5[i]=e22632[i]/(2*e34402[i]);
if e22617[i]=5 and e22632[i]>=0 and e34402[i]>0 then othpay5[i]=e22632[i]/(4.3*e34402[i]);

```

Appendix 2: Employment Variable Creation

```

if e22617[i]=6 and e22632[i]>=0 and e34402[i]>0 and e35600[i]>0 then othpay5[i]=e22632[i]/(e35600[i]*e34402[i]);
if e22617[i]=8 and e22632[i]>=0 and e34402[i]>0 then othpay5[i]=e22632[i]/(2.15*e34402[i]);
if e22617[i] in (12,13) and e22632[i]>=0 then do;
  /* condition based on the time unit to finish per othpay5. */
  if e22632c[i]=3 and e22632cb[i]>0 and e22632e[i]>=0 and e34402[i]>0 then
    othpay5[i]=(e22632[i]/e22632cb[i])*e22632e[i]/e34402[i];
  if e22632c[i]=4 and e22632cb[i]>0 and e34402[i]>0 then othpay5[i]=(e22632[i]/e22632cb[i])/e34402[i];
  if e22632c[i]=5 and e22632cb[i]>0 and e34402[i]>0 then othpay5[i]=(e22632[i]/e22632cb[i])/(e34402[i]*4.3);
  /* missing value */
  if e22632c[i] in (3,4,5) and -4<e34402[i]<=0 then othpay5[i]=-3;
end;
if e22617[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay5[i]=-3;
if e22617[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
  othpay5[i]=e34402[i];
if e22617[i]=6 and e35600[i] le 0 then othpay5[i]=-3;
if e22617[i]=6 and -4<e35600[i]<0 then othpay5[i]=e35600[i];
end;
end;

***** do it for non-overtime payment if answered in the later part *****/
***** for tips *****

/*report hourly wage at the beginning*/
do i=1 to 9;
if e102051[i]=1 and e102141[i]>=0 then othpay1[i]=e102141[i];
if e88000[i]>0 and e102051[i]=2 and e10214b1[i]>0 and e102141[i]>=0 then
  othpay1[i]=(e102141[i]*e10214b1[i])/e88000[i];
if e88000[i]>0 and e102051[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e102141[i]>=0
  then othpay1[i]=e102141[i]/e88000[i];
if e88000[i]>0 and e102051[i]=4 and e102141[i]>=0 then othpay1[i]=e102141[i]/(2*e88000[i]);
if e88000[i]>0 and e102051[i]=5 and e102141[i]>=0 then othpay1[i]=e102141[i]/(4.3*e88000[i]);
if e88000[i]>0 and e102051[i]=6 and e102141[i]>=0 and e99500[i]>0 then
  othpay1[i]=e102141[i]/(e99500[i]*e88000[i]);
if e88000[i]>0 and e102051[i]=8 and e102141[i]>=0 then othpay1[i]=e102141[i]/(2.15*e88000[i]);
if e102051[i] in (9,14) then othpay1[i]=0;
if e102051[i] in (12,13) and e102141[i]>=0 then do;
  /* condition based on the time unit to finish per othpay1. */
  if e10214aaa1[i]=1 and e10214aab1[i]>0 then othpay1[i]=(e102141[i]/e10214aab1[i])*60;
  if e10214aaa1[i]=2 and e10214aab1[i]>0 then othpay1[i]=e102141[i]/e10214aab1[i];
  if e10214aaa1[i]=3 and e10214aab1[i]>0 and e10214b1[i]>=0 and e88000[i]>0 then
    othpay1[i]=(e102141[i]/e10214aab1[i])*e10214b1[i]/e88000[i];
  if e10214aaa1[i]=4 and e10214aab1[i]>0 and e88000[i]>0 then othpay1[i]=(e102141[i]/e10214aab1[i])/e88000[i];
  if e10214aaa1[i]=5 and e10214aab1[i]>0 and e88000[i]>0 then
    othpay1[i]=(e102141[i]/e10214aab1[i])/(e88000[i]*4.3);
  /* missing value */
  if -4<e10214aaa1[i]<0 or -4<e10214aab1[i]<=0 then othpay1[i]=-3;
  if e10214aaa1[i] in (3,4,5) and -4<e88000[i]<=0 then othpay1[i]=-3;
  if e10214aaa1[i]=3 and e10214b1[i]<0 then othpay1[i]=-3;
end;
if e102051[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e102141[i]<0 then
  othpay1[i]=e102141[i];
if e102051[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e88000[i]=0 then othpay1[i]=-3;
if e102051[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e88000[i]<0 then
  othpay1[i]=e88000[i];
if e102051[i]=2 and e10214b1[i] le 0 then othpay1[i]=-3;
if e102051[i]=2 and -4<e10214b1[i]<0 then othpay1[i]=e10214b1[i];

```

```

end;

/* non-hourly wage at the beginning, with overtime*/
do i=1 to 9;
if e98429[i]>0 and e102051[i]=2 and e10214b1[i]>0 and e102141[i]>=0 then
    othpay1[i]=(e102141[i]*e10214b1[i])/e98429[i];
if e98429[i]>0 and e102051[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e102141[i]>=0
    then othpay1[i]=e102141[i]/e98429[i];
if e98429[i]>0 and e102051[i]=4 and e102141[i]>=0 then othpay1[i]=e102141[i]/(2*e98429[i]);
if e98429[i]>0 and e102051[i]=5 and e102141[i]>=0 then othpay1[i]=e102141[i]/(4.3*e98429[i]);
if e98429[i]>0 and e102051[i]=6 and e102141[i]>=0 and e99500[i]>0 then
    othpay1[i]=e102141[i]/(e99500[i]*e98429[i]);
if e98429[i]>0 and e102051[i]=8 and e102141[i]>=0 then othpay1[i]=e102141[i]/(2.15*e98429[i]);
if e102051[i] in (12,13) and e102141[i]>=0 then do;
    /* condition based on the time unit to finish per othpay1. */
    if e10214aaa1[i]=3 and e10214aab1[i]>0 and e10214b1[i]>=0 and e98429[i]>0 then
        othpay1[i]=(e102141[i]/e10214aab1[i])*e10214b1[i]/e98429[i];
    if e10214aaa1[i]=4 and e10214aab1[i]>0 and e98429[i]>0 then othpay1[i]=(e102141[i]/e10214aab1[i])/e98429[i];
    if e10214aaa1[i]=5 and e10214aab1[i]>0 and e98429[i]>0 then
        othpay1[i]=(e102141[i]/e10214aab1[i])/(e98429[i]*4.3);
    /* missing value */
    if e10214aaa1[i] in (3,4,5) and -4<e98429[i]<=0 then othpay1[i]=-3;
end;
if e102051[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999, -1, -2) and e98429[i]=0 then othpay1[i]=-3;
if e102051[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999, -1, -2) and -4<e98429[i]<0 then
    othpay1[i]=e98429[i];

end;

/* non-hourly wage at the beginning, without overtime*/
do i=1 to 9;
if e98402[i]>0 and e102051[i]=2 and e10214b1[i]>0 and e102141[i]>=0 then
    othpay1[i]=(e102141[i]*e10214b1[i])/e98402[i];
if e98402[i]>0 and e102051[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e102141[i]>=0
    then othpay1[i]=e102141[i]/e98402[i];
if e98402[i]>0 and e102051[i]=4 and e102141[i]>=0 then othpay1[i]=e102141[i]/(2*e98402[i]);
if e98402[i]>0 and e102051[i]=5 and e102141[i]>=0 then othpay1[i]=e102141[i]/(4.3*e98402[i]);
if e98402[i]>0 and e102051[i]=6 and e102141[i]>=0 and e99500[i]>0 then othpay1[i]=e102141[i]/(52*e98402[i]);
if e98402[i]>0 and e102051[i]=8 and e102141[i]>=0 then othpay1[i]=e102141[i]/(2.15*e98402[i]);
if e102051[i] in (7,9,12,13,15,16,17,14,18,19,21,22,23,24,25,26,28) then othpf1[i]=othpf1[i]+1;
if e102051[i] in (12,13) and e102141[i]>=0 then do;
    /* condition based on the time unit to finish per othpay1. */
    if e10214aaa1[i]=3 and e10214aab1[i]>0 and e10214b1[i]>=0 and e98402[i]>0 then
        othpay1[i]=(e102141[i]/e10214aab1[i])*e10214b1[i]/e98402[i];
    if e10214aaa1[i]=4 and e10214aab1[i]>0 and e98402[i]>0 then othpay1[i]=(e102141[i]/e10214aab1[i])/e98402[i];
    if e10214aaa1[i]=5 and e10214aab1[i]>0 and e98402[i]>0 then
        othpay1[i]=(e102141[i]/e10214aab1[i])/(e98402[i]*4.3);
    /* missing value */
    if e10214aaa1[i] in (3,4,5) and -4<e98402[i]<=0 then othpay1[i]=-3;
end;
if e102051[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999, -1, -2) and e98402[i]=0 then othpay1[i]=-3;
if e102051[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999, -1, -2) and -4<e98402[i]<0 then
    othpay1[i]=e98402[i];
if e102051[i]=6 and e99500[i] le 0 then othpay1[i]=-3;
if e102051[i]=6 and -4<e99500[i]<0 then othpay1[i]=e99500[i];
end;

```

Appendix 2: Employment Variable Creation

```

/* if tips is corrected in the later part */
do i=1 to 9;
if e102255[i]=1 then do;

/*report hourly wage at the beginning*/
if e100234[i]=1 and e100250[i]>=0 then othpay1[i]=e100250[i];
if e100234[i]=2 and e100250[i]>=0 and e100250e[i]>0 and e88000[i]>0 then
    othpay1[i]=e100250[i]*e100250e[i]/e88000[i];
if e100234[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e100250[i]>=0 and e88000[i]>0
    then othpay1[i]=e100250[i]/e88000[i];
if e100234[i]=4 and e100250[i]>=0 and e88000[i]>0 then othpay1[i]=e100250[i]/(2*e88000[i]);
if e100234[i]=5 and e100250[i]>=0 and e88000[i]>0 then othpay1[i]=e100250[i]/(4.3*e88000[i]);
if e100234[i]=6 and e100250[i]>=0 and e88000[i]>0 and e99500[i]>0 then
    othpay1[i]=e100250[i]/(e99500[i]*e88000[i]);
if e100234[i]=8 and e100250[i]>=0 and e88000[i]>0 then othpay1[i]=e100250[i]/(2.15*e88000[i]);
if e100234[i] in (9,14) then othpay1[i]=0;
if e100234[i] in (12,13) and e100250[i]>=0 then do;
    /* condition based on the time unit to finish per othpay1. */
    if e100250c[i]=1 and e100250cb[i]>0 then othpay1[i]=(e100250[i]/e100250cb[i])*60;
    if e100250c[i]=2 and e100250cb[i]>0 then othpay1[i]=e100250[i]/e100250cb[i];
    if e100250c[i]=3 and e100250cb[i]>0 and e100250e[i]>=0 and e88000[i]>0 then
        othpay1[i]=(e100250[i]/e100250cb[i])*e100250e[i]/e88000[i];
    if e100250c[i]=4 and e100250cb[i]>0 and e88000[i]>0 then othpay1[i]=(e100250[i]/e100250cb[i])/e88000[i];
    if e100250c[i]=5 and e100250cb[i]>0 and e88000[i]>0 then
        othpay1[i]=(e100250[i]/e100250cb[i])/(e88000[i]*4.3);
    /* missing value */
    if -4<e100250c[i]<0 or -4<e100250cb[i]<=0 then othpay1[i]=-3;
    if e100250c[i] in (3,4,5) and -4<e88000[i]<=0 then othpay1[i]=-3;
    if e100250c[i]=3 and e100250e[i]<0 then othpay1[i]=-3;
end;
if e100234[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e100250[i]<0 then
    othpay1[i]=e100250[i];
if e100234[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e88000[i]=0 then othpay1[i]=-3;
if e100234[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e88000[i]<0 then
    othpay1[i]=e88000[i];
if e100234[i]=2 and e100250e[i] le 0 then othpay1[i]=-3;
if e100234[i]=2 and -4<e100250e[i]<0 then othpay1[i]=e100250e[i];

/* non-hourly wage at the beginning, with overtime*/
if e100234[i]=2 and e100250[i]>=0 and e100250e[i]>0 and e98429[i]>0 then
    othpay1[i]=e100250[i]*e100250e[i]/e98429[i];
if e100234[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e100250[i]>=0 and e98429[i]>0
    then othpay1[i]=e100250[i]/e98429[i];
if e100234[i]=4 and e100250[i]>=0 and e98429[i]>0 then othpay1[i]=e100250[i]/(2*e98429[i]);
if e100234[i]=5 and e100250[i]>=0 and e98429[i]>0 then othpay1[i]=e100250[i]/(4.3*e98429[i]);
if e100234[i]=6 and e100250[i]>=0 and e98429[i]>0 and e99500[i]>0 then
    othpay1[i]=e100250[i]/(e99500[i]*e98429[i]);
if e100234[i]=8 and e100250[i]>=0 and e98429[i]>0 then othpay1[i]=e100250[i]/(2.15*e98429[i]);
if e100234[i] in (12,13) and e100250[i]>=0 then do;
    /* condition based on the time unit to finish per othpay1. */
    if e100250c[i]=3 and e100250cb[i]>0 and e100250e[i]>=0 and e98429[i]>0 then
        othpay1[i]=(e100250[i]/e100250cb[i])*e100250e[i]/e98429[i];
    if e100250c[i]=4 and e100250cb[i]>0 and e98429[i]>0 then othpay1[i]=(e100250[i]/e100250cb[i])/e98429[i];
    if e100250c[i]=5 and e100250cb[i]>0 and e98429[i]>0 then
        othpay1[i]=(e100250[i]/e100250cb[i])/(e98429[i]*4.3);
    /* missing value */
    if e100250c[i] in (3,4,5) and -4<e98429[i]<=0 then othpay1[i]=-3;

```

```

end;
if e100234[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e98429[i]=0 then othpay1[i]=-3;
if e100234[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e98429[i]<0 then
    othpay1[i]=e98429[i];

/* non-hourly wage at the beginning, without overtime*/
if e100234[i]=2 and e100250[i]>=0 and e100250e[i]>0 and e98402[i]>0 then
    othpay1[i]=e100250[i]*e100250e[i]/e98402[i];
if e100234[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e100250[i]>=0 and e98402[i]>0
    then othpay1[i]=e100250[i]/e98402[i];
if e100234[i]=4 and e100250[i]>=0 and e98402[i]>0 then othpay1[i]=e100250[i]/(2*e98402[i]);
if e100234[i]=5 and e100250[i]>=0 and e98402[i]>0 then othpay1[i]=e100250[i]/(4.3*e98402[i]);
if e100234[i]=6 and e100250[i]>=0 and e98402[i]>0 and e99500[i]>0 then
    othpay1[i]=e100250[i]/(e99500[i]*e98402[i]);
if e100234[i]=8 and e100250[i]>=0 and e98402[i]>0 then othpay1[i]=e100250[i]/(2.15*e98402[i]);
    if e100234[i] in (12,13) and e100250[i]>=0 then do;
        /* condition based on the time unit to finish per othpay1. */
        if e100250c[i]=3 and e100250cb[i]>0 and e100250e[i]>=0 and e98402[i]>0 then
            othpay1[i]=(e100250[i]/e100250cb[i])*e100250e[i]/e98402[i];
        if e100250c[i]=4 and e100250cb[i]>0 and e98402[i]>0 then othpay1[i]=(e100250[i]/e100250cb[i])/e98402[i];
        if e100250c[i]=5 and e100250cb[i]>0 and e98402[i]>0 then
            othpay1[i]=(e100250[i]/e100250cb[i])/(e98402[i]*4.3);
        /* missing value */
        if e100250c[i] in (3,4,5) and -4<e98402[i]<=0 then othpay1[i]=-3;
    end;
    if e100234[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e98402[i]=0 then othpay1[i]=-3;
    if e100234[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e98402[i]<0 then
        othpay1[i]=e98402[i];

if e100234[i]=6 and e99500[i] le 0 then othpay1[i]=-3;
if e100234[i]=6 and -4<e99500[i]<0 then othpay1[i]=e99500[i];

end;
end;

***** for commissions *****

/*report hourly wage at the beginning*/
do i=1 to 9;
if e102052[i]=1 and e102142[i]>=0 then othpay2[i]=e102142[i];
if e88000[i]>0 and e102052[i]=2 and e10214b2[i]>0 and e102142[i]>=0 then
    othpay2[i]=(e102142[i]*e10214b2[i])/e88000[i];
if e88000[i]>0 and e102052[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e102142[i]>=0
    then othpay2[i]=e102142[i]/e88000[i];
if e88000[i]>0 and e102052[i]=4 and e102142[i]>=0 then othpay2[i]=e102142[i]/(2*e88000[i]);
if e88000[i]>0 and e102052[i]=5 and e102142[i]>=0 then othpay2[i]=e102142[i]/(4.3*e88000[i]);
if e88000[i]>0 and e102052[i]=6 and e102142[i]>=0 and e99500[i]>0 then
    othpay2[i]=(e102142[i]/(e99500[i]*e88000[i]));
if e88000[i]>0 and e102052[i]=8 and e102142[i]>=0 then othpay2[i]=e102142[i]/(2.15*e88000[i]);
if e102052[i] in (9,14) then othpay2[i]=0;
if e102052[i] in (12,13) and e102142[i]>=0 then do;
    /* condition based on the time unit to finish per othpay2. */
    if e10214aaa2[i]=1 and e10214aab2[i]>0 then othpay2[i]=(e102142[i]/e10214aab2[i])*60;
    if e10214aaa2[i]=2 and e10214aab2[i]>0 then othpay2[i]=e102142[i]/e10214aab2[i];
    if e10214aaa2[i]=3 and e10214aab2[i]>0 and e10214b2[i]>=0 and e88000[i]>0 then
        othpay2[i]=(e102142[i]/e10214aab2[i])*e10214b2[i]/e88000[i];
    if e10214aaa2[i]=4 and e10214aab2[i]>0 and e88000[i]>0 then othpay2[i]=(e102142[i]/e10214aab2[i])/e88000[i];

```

```

if e10214aaa2[i]=5 and e10214aab2[i]>0 and e88000[i]>0 then
    othpay2[i]=(e102142[i]/e10214aab2[i])/(e88000[i]*4.3);
/* missing value */
if -4<e10214aaa2[i]<0 or -4<e10214aab2[i]<=0 then othpay2[i]=-3;
if e10214aaa2[i] in (3,4,5) and -4<e88000[i]<=0 then othpay2[i]=-3;
if e10214aaa2[i]=3 and e10214b2[i]<0 then othpay2[i]=-3;
end;
if e102052[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e102142[i]<0 then
    othpay2[i]=e102142[i];
if e102052[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e88000[i]=0 then othpay2[i]=-3;
if e102052[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e88000[i]<0 then
    othpay2[i]=e88000[i];
if e102052[i]=2 and e10214b2[i] le 0 then othpay2[i]=-3;
if e102052[i]=2 and -4<e10214b2[i]<0 then othpay2[i]=e10214b2[i];
end;

/* non-hourly wage at the beginning, with overtime*/
do i=1 to 9;
if e98429[i]>0 and e102052[i]=2 and e10214b2[i]>0 and e102142[i]>=0 then
    othpay2[i]=(e102142[i]*e10214b2[i])/e98429[i];
if e98429[i]>0 and e102052[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e102142[i]>=0
    then othpay2[i]=e102142[i]/e98429[i];
if e98429[i]>0 and e102052[i]=4 and e102142[i]>=0 then othpay2[i]=e102142[i]/(2*e98429[i]);
if e98429[i]>0 and e102052[i]=5 and e102142[i]>=0 then othpay2[i]=e102142[i]/(4.3*e98429[i]);
if e98429[i]>0 and e102052[i]=6 and e102142[i]>=0 and e99500[i]>0 then
    othpay2[i]=e102142[i]/(e99500[i]*e98429[i]);
if e98429[i]>0 and e102052[i]=8 and e102142[i]>=0 then othpay2[i]=e102142[i]/(2.15*e98429[i]);
if e102052[i] in (12,13) and e102142[i]>=0 then do;
    /* condition based on the time unit to finish per othpay2. */
    if e10214aaa2[i]=3 and e10214aab2[i]>0 and e10214b2[i]>=0 and e98429[i]>0 then
        othpay2[i]=(e102142[i]/e10214aab2[i])*e10214b2[i]/e98429[i];
    if e10214aaa2[i]=4 and e10214aab2[i]>0 and e98429[i]>0 then othpay2[i]=(e102142[i]/e10214aab2[i])/e98429[i];
    if e10214aaa2[i]=5 and e10214aab2[i]>0 and e98429[i]>0 then
        othpay2[i]=(e102142[i]/e10214aab2[i])/(e98429[i]*4.3);
    /* missing value */
    if e10214aaa2[i] in (3,4,5) and -4<e98429[i]<=0 then othpay2[i]=-3;
end;
if e102052[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e98429[i]=0 then othpay2[i]=-3;
if e102052[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e98429[i]<0 then
    othpay2[i]=e98429[i];
end;

/* non-hourly wage at the beginning, without overtime*/
do i=1 to 9;
if e98402[i]>0 and e102052[i]=2 and e10214b2[i]>0 and e102142[i]>=0 then
    othpay2[i]=(e102142[i]*e10214b2[i])/e98402[i];
if e98402[i]>0 and e102052[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e102142[i]>=0
    then othpay2[i]=e102142[i]/e98402[i];
if e98402[i]>0 and e102052[i]=4 and e102142[i]>=0 then othpay2[i]=e102142[i]/(2*e98402[i]);
if e98402[i]>0 and e102052[i]=5 and e102142[i]>=0 then othpay2[i]=e102142[i]/(4.3*e98402[i]);
if e98402[i]>0 and e102052[i]=6 and e102142[i]>=0 and e99500[i]>0 then
    othpay2[i]=e102142[i]/(e99500[i]*e98402[i]);
if e98402[i]>0 and e102052[i]=8 and e102142[i]>=0 then othpay2[i]=e102142[i]/(2.15*e98402[i]);
if e102052[i] in (7,9,12,13,15,16,17,14,18,19,21,22,23,24,25,26,28) then othpf2[i]=othpf2[i]+1;
if e102052[i] in (12,13) and e102142[i]>=0 then do;

```

```

/* condition based on the time unit to finish per othpay2. */
if e10214aaa2[i]=3 and e10214aab2[i]>0 and e10214b2[i]>=0 and e98402[i]>0 then
    othpay2[i]=(e102142[i]/e10214aab2[i])*e10214b2[i]/e98402[i];
if e10214aaa2[i]=4 and e10214aab2[i]>0 and e98402[i]>0 then othpay2[i]=(e102142[i]/e10214aab2[i])/e98402[i];
if e10214aaa2[i]=5 and e10214aab2[i]>0 and e98402[i]>0 then
    othpay2[i]=(e102142[i]/e10214aab2[i])/(e98402[i]*4.3);
/* missing value */
if e10214aaa2[i] in (3,4,5) and -4<e98402[i]<=0 then othpay2[i]=-3;
end;
if e102052[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e98402[i]=0 then othpay2[i]=-3;
if e102052[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e98402[i]<0 then
    othpay2[i]=e98402[i];
if e102052[i]=6 and e99500[i] le 0 then othpay2[i]=-3;
if e102052[i]=6 and -4<e99500[i]<0 then othpay2[i]=e99500[i];
end;

/* if commissions is corrected in the later part */
do i=1 to 9;
if e102256[i]=1 then do;
/*report hourly wage at the beginning*/
if e100235[i]=1 and e100251[i]>=0 then othpay2[i]=e100251[i];
if e100235[i]=2 and e100251[i]>=0 and e100251e[i]>0 and e88000[i]>0 then
    othpay2[i]=e100251[i]*e100251e[i]/e88000[i];
if e100235[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e100251[i]>=0 and e88000[i]>0
    then othpay2[i]=e100251[i]/e88000[i];
if e100235[i]=4 and e100251[i]>=0 and e88000[i]>0 then othpay2[i]=e100251[i]/(2*e88000[i]);
if e100235[i]=5 and e100251[i]>=0 and e88000[i]>0 then othpay2[i]=e100251[i]/(4.3*e88000[i]);
if e100235[i]=6 and e100251[i]>=0 and e88000[i]>0 and e99500[i]>0 then
    othpay2[i]=e100251[i]/(e99500[i]*e88000[i]);
if e100235[i]=8 and e100251[i]>=0 and e88000[i]>0 then othpay2[i]=e100251[i]/(2.15*e88000[i]);
if e100235[i] in (9,14) then othpay2[i]=0;
if e100235[i] in (12,13) and e100251[i]>=0 then do;
    /* condition based on the time unit to finish per othpay2. */
    if e100251c[i]=1 and e100251cb[i]>0 then othpay2[i]=(e100251[i]/e100251cb[i])*60;
    if e100251c[i]=2 and e100251cb[i]>0 then othpay2[i]=e100251[i]/e100251cb[i];
    if e100251c[i]=3 and e100251cb[i]>0 and e100251e[i]>=0 and e88000[i]>0 then
        othpay2[i]=(e100251[i]/e100251cb[i])*e100251e[i]/e88000[i];
    if e100251c[i]=4 and e100251cb[i]>0 and e88000[i]>0 then othpay2[i]=(e100251[i]/e100251cb[i])/e88000[i];
    if e100251c[i]=5 and e100251cb[i]>0 and e88000[i]>0 then
        othpay2[i]=(e100251[i]/e100251cb[i])/(e88000[i]*4.3);
/* missing value */
    if -4<e100251c[i]<0 or -4<e100251cb[i]<=0 then othpay2[i]=-3;
    if e100251c[i] in (3,4,5) and -4<e88000[i]<=0 then othpay2[i]=-3;
    if e100251c[i]=3 and e100251e[i]<0 then othpay2[i]=-3;
end;
if e100235[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e100251[i]<0 then
    othpay2[i]=e100251[i];
if e100235[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e88000[i]=0 then othpay2[i]=-3;
if e100235[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e88000[i]<0 then
    othpay2[i]=e88000[i];
if e100235[i]=2 and e100251e[i] le 0 then othpay2[i]=-3;
if e100235[i]=2 and -4<e100251e[i]<0 then othpay2[i]=e100251e[i];

/* non-hourly wage at the beginning, with overtime*/
if e100235[i]=2 and e100251[i]>=0 and e100251e[i]>0 and e98429[i]>0 then
    othpay2[i]=e100251[i]*e100251e[i]/e98429[i];

```

Appendix 2: Employment Variable Creation

```

if e100235[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e100251[i]>=0 and e98429[i]>0
    then othpay2[i]=e100251[i]/e98429[i];
if e100235[i]=4 and e100251[i]>=0 and e98429[i]>0 then othpay2[i]=e100251[i]/(2*e98429[i]);
if e100235[i]=5 and e100251[i]>=0 and e98429[i]>0 then othpay2[i]=e100251[i]/(4.3*e98429[i]);
if e100235[i]=6 and e100251[i]>=0 and e98429[i]>0 and e99500[i]>0 then
    othpay2[i]=e100251[i]/(e99500[i]*e98429[i]);
if e100235[i]=8 and e100251[i]>=0 and e98429[i]>0 then othpay2[i]=e100251[i]/(2.15*e98429[i]);
if e100235[i] in (12, 13) and e100251[i]>=0 then do;
    /* condition based on the time unit to finish per othpay2. */
    if e100251c[i]=3 and e100251cb[i]>0 and e100251e[i]>=0 and e98429[i]>0 then
        othpay2[i]=(e100251[i]/e100251cb[i])*e100251e[i]/e98429[i];
    if e100251c[i]=4 and e100251cb[i]>0 and e98429[i]>0 then othpay2[i]=(e100251[i]/e100251cb[i])/e98429[i];
    if e100251c[i]=5 and e100251cb[i]>0 and e98429[i]>0 then
        othpay2[i]=(e100251[i]/e100251cb[i])/(e98429[i]*4.3);
    /* missing value */
    if e100251c[i] in (3, 4, 5) and -4<e98429[i]<=0 then othpay2[i]=-3;
end;
if e100235[i] in (2, 3, 4, 5, 6, 7, 8, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e98429[i]=0 then othpay2[i]=-3;
if e100235[i] in (2, 3, 4, 5, 6, 7, 8, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and -4<e98429[i]<0 then
    othpay2[i]=e98429[i];

/* non-hourly wage at the beginning, without overtime*/
if e100235[i]=2 and e100251[i]>=0 and e100251e[i]>0 and e98402[i]>0 then
    othpay2[i]=e100251[i]*e100251e[i]/e98402[i];
if e100235[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e100251[i]>=0 and e98402[i]>0
    then othpay2[i]=e100251[i]/e98402[i];
if e100235[i]=4 and e100251[i]>=0 and e98402[i]>0 then othpay2[i]=e100251[i]/(2*e98402[i]);
if e100235[i]=5 and e100251[i]>=0 and e98402[i]>0 then othpay2[i]=e100251[i]/(4.3*e98402[i]);
if e100235[i]=6 and e100251[i]>=0 and e98402[i]>0 and e99500[i]>0 then
    othpay2[i]=e100251[i]/(e99500[i]*e98402[i]);
if e100235[i]=8 and e100251[i]>=0 and e98402[i]>0 then othpay2[i]=e100251[i]/(2.15*e98402[i]);
if e100235[i] in (12, 13) and e100251[i]>=0 then do;
    /* condition based on the time unit to finish per othpay2. */
    if e100251c[i]=3 and e100251cb[i]>0 and e100251e[i]>=0 and e98402[i]>0 then
        othpay2[i]=(e100251[i]/e100251cb[i])*e100251e[i]/e98402[i];
    if e100251c[i]=4 and e100251cb[i]>0 and e98402[i]>0 then othpay2[i]=(e100251[i]/e100251cb[i])/e98402[i];
    if e100251c[i]=5 and e100251cb[i]>0 and e98402[i]>0 then
        othpay2[i]=(e100251[i]/e100251cb[i])/(e98402[i]*4.3);
    /* missing value */
    if e100251c[i] in (3, 4, 5) and -4<e98402[i]<=0 then othpay2[i]=-3;
end;
if e100235[i] in (2, 3, 4, 5, 6, 7, 8, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e98402[i]=0 then othpay2[i]=-3;
if e100235[i] in (2, 3, 4, 5, 6, 7, 8, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and -4<e98402[i]<0 then
    othpay2[i]=e98402[i];
if e100235[i]=6 and e99500[i]>0 then othpay2[i]=-3;
if e100235[i]=6 and -4<e99500[i]<0 then othpay2[i]=e99500[i];
end;
end;

***** for bonuses *****

/*report hourly wage at the beginning*/
do i=1 to 9;
if e102053[i]=1 and e102143[i]>=0 then othpay3[i]=e102143[i];
if e88000[i]>0 and e102053[i]=2 and e10214b3[i]>0 and e102143[i]>=0 then
    othpay3[i]=(e102143[i]*e10214b3[i])/e88000[i];

```

```

if e88000[i]>0 and e102053[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e102143[i]>=0
    then othpay3[i]=e102143[i]/e88000[i];
if e88000[i]>0 and e102053[i]=4 and e102143[i]>=0 then othpay3[i]=e102143[i]/(2*e88000[i]);
if e88000[i]>0 and e102053[i]=5 and e102143[i]>=0 then othpay3[i]=e102143[i]/(4.3*e88000[i]);
if e88000[i]>0 and e102053[i]=6 and e102143[i]>=0 and e99500[i]>0 then
    othpay3[i]=e102143[i]/(e99500[i]*e88000[i]);
if e88000[i]>0 and e102053[i]=8 and e102143[i]>=0 then othpay3[i]=e102143[i]/(2.15*e88000[i]);
if e102053[i] in (9,14) then othpay2[i]=0;
if e102053[i] in (12,13) and e102143[i]>=0 then do;
    /* condition based on the time unit to finish per othpay2. */
    if e10214aaa3[i]=1 and e10214aab3[i]>0 then othpay3[i]=(e102143[i]/e10214aab3[i])*60;
    if e10214aaa3[i]=2 and e10214aab3[i]>0 then othpay3[i]=e102143[i]/e10214aab3[i];
    if e10214aaa3[i]=3 and e10214aab3[i]>0 and e10214b3[i]>=0 and e88000[i]>0 then
        othpay3[i]=(e102143[i]/e10214aab3[i])*e10214b3[i]/e88000[i];
    if e10214aaa3[i]=4 and e10214aab3[i]>0 and e88000[i]>0 then othpay3[i]=(e102143[i]/e10214aab3[i])/e88000[i];
    if e10214aaa3[i]=5 and e10214aab3[i]>0 and e88000[i]>0 then
        othpay3[i]=(e102143[i]/e10214aab3[i])/(e88000[i]*4.3);
    /* missing value */
    if -4<e10214aaa3[i]<0 or -4<e10214aab3[i]<=0 then othpay3[i]=-3;
    if e10214aaa3[i] in (3,4,5) and -4<e88000[i]<=0 then othpay3[i]=-3;
    if e10214aaa3[i]=3 and e10214b3[i]<0 then othpay3[i]=-3;
end;
if e102053[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e102143[i]<0 then
    othpay3[i]=e102143[i];
if e102053[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e88000[i]=0 then othpay3[i]=-3;
if e102053[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e88000[i]<0 then
    othpay3[i]=e88000[i];
if e102053[i]=2 and e10214b3[i] le 0 then othpay3[i]=-3;
if e102053[i]=2 and -4<e10214b3[i]<0 then othpay3[i]=e10214b3[i];
end;

/* non-hourly wage at the beginning, with overtime*/
do i=1 to 9;
if e98429[i]>0 and e102053[i]=2 and e10214b3[i]>0 and e102143[i]>=0 then
    othpay3[i]=(e102143[i]*e10214b3[i])/e98429[i];
if e98429[i]>0 and e102053[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e102143[i]>=0
    then othpay3[i]=e102143[i]/e98429[i];
if e98429[i]>0 and e102053[i]=4 and e102143[i]>=0 then othpay3[i]=e102143[i]/(2*e98429[i]);
if e98429[i]>0 and e102053[i]=5 and e102143[i]>=0 then othpay3[i]=e102143[i]/(4.3*e98429[i]);
if e98429[i]>0 and e102053[i]=6 and e102143[i]>=0 and e99500[i]>0 then
    othpay3[i]=e102143[i]/(e99500[i]*e98429[i]);
if e98429[i]>0 and e102053[i]=8 and e102143[i]>=0 then othpay3[i]=e102143[i]/(2.15*e98429[i]);
if e102053[i] in (12,13) and e102143[i]>=0 then do;
    /* condition based on the time unit to finish per othpay3. */
    if e10214aaa3[i]=3 and e10214aab3[i]>0 and e10214b3[i]>=0 and e98429[i]>0 then
        othpay3[i]=(e102143[i]/e10214aab3[i])*e10214b3[i]/e98429[i];
    if e10214aaa3[i]=4 and e10214aab3[i]>0 and e98429[i]>0 then othpay3[i]=(e102143[i]/e10214aab3[i])/e98429[i];
    if e10214aaa3[i]=5 and e10214aab3[i]>0 and e98429[i]>0 then
        othpay3[i]=(e102143[i]/e10214aab3[i])/(e98429[i]*4.3);
    /* missing value */
    if e10214aaa3[i] in (3,4,5) and -4<e98429[i]<=0 then othpay3[i]=-3;
end;
if e102053[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e98429[i]=0 then othpay3[i]=-3;
if e102053[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e98429[i]<0 then
    othpay3[i]=e98429[i];
end;

```

```

/* non-hourly wage at the beginning, without overtime*/
do i=1 to 9;
if e98402[i]>0 and e102053[i]=2 and e10214b3[i]>0 and e102143[i]>=0 then
    othpay3[i]=(e102143[i]*e10214b3[i])/e98402[i];
if e98402[i]>0 and e102053[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e102143[i]>=0
    then othpay3[i]=e102143[i]/e98402[i];
if e98402[i]>0 and e102053[i]=4 and e102143[i]>=0 then othpay3[i]=e102143[i]/(2*e98402[i]);
if e98402[i]>0 and e102053[i]=5 and e102143[i]>=0 then othpay3[i]=e102143[i]/(4.3*e98402[i]);
if e98402[i]>0 and e102053[i]=6 and e102143[i]>=0 and e99500[i]>0 then
    othpay3[i]=e102143[i]/(e99500[i]*e98402[i]);
if e98402[i]>0 and e102053[i]=8 and e102143[i]>=0 then othpay3[i]=e102143[i]/(2.15*e98402[i]);
if e102053[i] in (7,9,12,13,15,16,17,14,18,19,21,22,23,24,25,26,28) then othpf3[i]=othpf3[i]+1;
if e102053[i] in (12,13) and e102143[i]>=0 then do;
    /* condition based on the time unit to finish per othpay3. */
    if e10214aaa3[i]=3 and e10214aab3[i]>0 and e10214b3[i]>=0 and e98402[i]>0 then
        othpay3[i]=(e102143[i]/e10214aab3[i])*e10214b3[i]/e98402[i];
    if e10214aaa3[i]=4 and e10214aab3[i]>0 and e98402[i]>0 then othpay3[i]=(e102143[i]/e10214aab3[i])/e98402[i];
    if e10214aaa3[i]=5 and e10214aab3[i]>0 and e98402[i]>0 then
        othpay3[i]=(e102143[i]/e10214aab3[i])/(e98402[i]*4.3);
    /* missing value */
    if e10214aaa3[i] in (3,4,5) and -4<e98402[i]<=0 then othpay3[i]=-3;
end;
if e102053[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e98402[i]=0 then othpay3[i]=-3;
if e102053[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e98402[i]<0 then
    othpay3[i]=e98402[i];
if e102053[i]=6 and e99500[i] le 0 then othpay3[i]=-3;
if e102053[i]=6 and -4<e99500[i]<0 then othpay3[i]=e99500[i];
if e102053[i]=6 and e99500[i]>0 then othpay3[i]=e99500[i];
end;

/* if bonus is corrected in the later part */
do i=1 to 9;
if e102257[i]=1 then do;
    /*report hourly wage at the beginning*/
    if e100236[i]=1 and e100252[i]>=0 then othpay3[i]=e100252[i];
    if e100236[i]=2 and e100252[i]>=0 and e100252e[i]>0 and e88000[i]>0 then
        othpay3[i]=e100252[i]*e100252e[i]/e88000[i];
    if e100236[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e100252[i]>=0 and e88000[i]>0
        then othpay3[i]=e100252[i]/e88000[i];
    if e100236[i]=4 and e100252[i]>=0 and e88000[i]>0 then othpay3[i]=e100252[i]/(2*e88000[i]);
    if e100236[i]=5 and e100252[i]>=0 and e88000[i]>0 then othpay3[i]=e100252[i]/(4.3*e88000[i]);
    if e100236[i]=6 and e100252[i]>=0 and e88000[i]>0 and e99500[i]>0 then
        othpay3[i]=(e100252[i]/e99500[i])*e88000[i];
    if e100236[i]=8 and e100252[i]>=0 and e88000[i]>0 then othpay3[i]=e100252[i]/(2.15*e88000[i]);
    if e100236[i] in (9,14) then othpay3[i]=0;
    if e100236[i] in (12,13) and e100252[i]>=0 then do;
        /* condition based on the time unit to finish per othpay3. */
        if e100252c[i]=1 and e100252cb[i]>0 then othpay3[i]=(e100252[i]/e100252cb[i])*60;
        if e100252c[i]=2 and e100252cb[i]>0 then othpay3[i]=(e100252[i]/e100252cb[i])*60;
        if e100252c[i]=3 and e100252cb[i]>0 and e100252e[i]>=0 and e88000[i]>0 then
            othpay3[i]=(e100252[i]/e100252cb[i])*e100252e[i]/e88000[i];
        if e100252c[i]=4 and e100252cb[i]>0 and e88000[i]>0 then othpay3[i]=(e100252[i]/e100252cb[i])/e88000[i];
        if e100252c[i]=5 and e100252cb[i]>0 and e88000[i]>0 then
            othpay3[i]=(e100252[i]/e100252cb[i])/(e88000[i]*4.3);
    /* missing value */
    if -4<e100252c[i]<0 or -4<e100252cb[i]<=0 then othpay3[i]=-3;

```

```

if e100252c[i] in (3,4,5) and -4<e88000[i]<=0 then othpay3[i]=-3;
  if e100252c[i]=3 and e100252e[i]<0 then othpay3[i]=-3;
end;
if e100236[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e100252[i]<0 then
  othpay3[i]=e100252[i];
if e100236[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e88000[i]=0 then othpay3[i]=-3;
if e100236[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e88000[i]<0 then
  othpay3[i]=e88000[i];
if e100236[i]=2 and e100252e[i] le 0 then othpay3[i]=-3;
if e100236[i]=2 and -4<e100252e[i]<0 then othpay3[i]=e100252e[i];

/* non-hourly wage at the beginning, with overtime*/
if e100236[i]=2 and e100252[i]>=0 and e100252e[i]>0 and e98429[i]>0 then
  othpay3[i]=e100252[i]*e100252e[i]/e98429[i];
if e100236[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e100252[i]>=0 and e98429[i]>0
  then othpay3[i]=e100252[i]/e98429[i];
if e100236[i]=4 and e100252[i]>=0 and e98429[i]>0 then othpay3[i]=e100252[i]/(2*e98429[i]);
if e100236[i]=5 and e100252[i]>=0 and e98429[i]>0 then othpay3[i]=e100252[i]/(4.3*e98429[i]);
if e100236[i]=6 and e100252[i]>=0 and e98429[i]>0 and e99500[i]>0 then
  othpay3[i]=e100252[i]/(e99500[i]*e98429[i]);
if e100236[i]=8 and e100252[i]>=0 and e98429[i]>0 then othpay3[i]=e100252[i]/(2.15*e98429[i]);
if e100236[i] in (12,13) and e100252[i]>=0 then do;
  /* condition based on the time unit to finish per othpay3. */
  if e100252c[i]=3 and e100252cb[i]>0 and e100252e[i]>=0 and e98429[i]>0 then
    othpay3[i]=(e100252[i]/e100252cb[i])*e100252e[i]/e98429[i];
  if e100252c[i]=4 and e100252cb[i]>0 and e98429[i]>0 then othpay3[i]=(e100252[i]/e100252cb[i])/e98429[i];
  if e100252c[i]=5 and e100252cb[i]>0 and e98429[i]>0 then
    othpay3[i]=(e100252[i]/e100252cb[i])/(e98429[i]*4.3);
  /* missing value */
  if e100252c[i] in (3,4,5) and -4<e98429[i]<=0 then othpay3[i]=-3;
end;
if e100236[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e98429[i]=0 then othpay3[i]=-3;
if e100236[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e98429[i]<0 then
  othpay3[i]=e98429[i];

/* non-hourly wage at the beginning, without overtime*/
if e100236[i]=2 and e100252[i]>=0 and e100252e[i]>0 and e98402[i]>0 then
  othpay3[i]=e100252[i]*e100252e[i]/e98402[i];
if e100236[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e100252[i]>=0 and e98402[i]>0
  then othpay3[i]=e100252[i]/e98402[i];
if e100236[i]=4 and e100252[i]>=0 and e98402[i]>0 then othpay3[i]=e100252[i]/(2*e98402[i]);
if e100236[i]=5 and e100252[i]>=0 and e98402[i]>0 then othpay3[i]=e100252[i]/(4.3*e98402[i]);
if e100236[i]=6 and e100252[i]>=0 and e98402[i]>0 and e99500[i]>0 then
  othpay3[i]=e100252[i]/(e99500[i]*e98402[i]);
if e100236[i]=8 and e100252[i]>=0 and e98402[i]>0 then othpay3[i]=e100252[i]/(2.15*e98402[i]);
if e100236[i] in (12,13) and e100252[i]>=0 then do;
  /* condition based on the time unit to finish per othpay3. */
  if e100252c[i]=3 and e100252cb[i]>0 and e100252e[i]>=0 and e98402[i]>0 then
    othpay3[i]=(e100252[i]/e100252cb[i])*e100252e[i]/e98402[i];
  if e100252c[i]=4 and e100252cb[i]>0 and e98402[i]>0 then othpay3[i]=(e100252[i]/e100252cb[i])/e98402[i];
  if e100252c[i]=5 and e100252cb[i]>0 and e98402[i]>0 then
    othpay3[i]=(e100252[i]/e100252cb[i])/(e98402[i]*4.3);
  /* missing value */
  if e100252c[i] in (3,4,5) and -4<e98402[i]<=0 then othpay3[i]=-3;
end;
if e100236[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e98402[i]=0 then othpay3[i]=-3;

```

Appendix 2: Employment Variable Creation

```

if e100236[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e98402[i]<0 then
    othpay3[i]=e98402[i];
if e100236[i]=6 and e99500[i] le 0 then othpay3[i]=-3;
if e100236[i]=6 and -4<e99500[i]<0 then othpay3[i]=e99500[i];
end;
end;

/***** for incentive pay *****/

/*report hourly wage at the beginning*/
do i=1 to 9;
if e102054[i]=1 and e102144[i]>=0 then othpay4[i]=e102144[i];
if e88000[i]>0 and e102054[i]=2 and e10214b4[i]>0 and e102144[i]>=0 then
    othpay4[i]=(e102144[i]*e10214b4[i])/e88000[i];
if e88000[i]>0 and e102054[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e102144[i]>=0
    then othpay4[i]=e102144[i]/e88000[i];
if e88000[i]>0 and e102054[i]=4 and e102144[i]>=0 then othpay4[i]=e102144[i]/(2*e88000[i]);
if e88000[i]>0 and e102054[i]=5 and e102144[i]>=0 then othpay4[i]=e102144[i]/(4.3*e88000[i]);
if e88000[i]>0 and e102054[i]=6 and e102144[i]>=0 and e99500[i]>0 then
    othpay4[i]=e102144[i]/(e99500[i]*e88000[i]);
if e88000[i]>0 and e102054[i]=8 and e102144[i]>=0 then othpay4[i]=e102144[i]/(2.15*e88000[i]);
if e102054[i] in (9, 14) then othpay4[i]=0;
if e102054[i] in (12,13) and e102144[i]>=0 then do;
    /* condition based on the time unit to finish per othpay2. */
    if e10214aaa4[i]=1 and e10214aab4[i]>0 then othpay4[i]=(e102144[i]/e10214aab4[i])*60;
    if e10214aaa4[i]=2 and e10214aab4[i]>0 then othpay4[i]=e102144[i]/e10214aab4[i];
    if e10214aaa4[i]=3 and e10214aab4[i]>0 and e10214b4[i]>=0 and e88000[i]>0 then
        othpay4[i]=(e102144[i]/e10214aab4[i])*e10214b4[i]/e88000[i];
    if e10214aaa4[i]=4 and e10214aab4[i]>0 and e88000[i]>0 then othpay4[i]=(e102144[i]/e10214aab4[i])/e88000[i];
    if e10214aaa4[i]=5 and e10214aab4[i]>0 and e88000[i]>0 then
        othpay4[i]=(e102144[i]/e10214aab4[i])/(e88000[i]*4.3);
    /* missing value */
    if -4<e10214aaa4[i]<0 or -4<e10214aab4[i]<=0 then othpay4[i]=-3;
    if e10214aaa4[i] in (3,4,5) and -4<e88000[i]<=0 then othpay4[i]=-3;
    if e10214aaa4[i]=3 and e10214b4[i]<0 then othpay4[i]=-3;
end;
if e102054[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e102144[i]<0 then
    othpay4[i]=e102144[i];
if e102054[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e88000[i]=0 then othpay4[i]=-3;
if e102054[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e88000[i]<0 then
    othpay4[i]=e88000[i];
if e102054[i]=2 and e10214b4[i] le 0 then othpay4[i]=-3;
if e102054[i]=2 and -4<e10214b4[i]<0 then othpay4[i]=e10214b4[i];

end;

/* non-hourly wage at the beginning, with overtime*/
do i=1 to 9;
if e98429[i]>0 and e102054[i]=2 and e10214b4[i]>0 and e102144[i]>=0 then
    othpay4[i]=(e102144[i]*e10214b4[i])/e98429[i];
if e98429[i]>0 and e102054[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e102144[i]>=0
    then othpay4[i]=e102144[i]/e98429[i];
if e98429[i]>0 and e102054[i]=4 and e102144[i]>=0 then othpay4[i]=e102144[i]/(2*e98429[i]);
if e98429[i]>0 and e102054[i]=5 and e102144[i]>=0 then othpay4[i]=e102144[i]/(4.3*e98429[i]);
if e98429[i]>0 and e102054[i]=6 and e102144[i]>=0 and e99500[i]>0 then
    othpay4[i]=e102144[i]/(e99500[i]*e98429[i]);
if e98429[i]>0 and e102054[i]=8 and e102144[i]>=0 then othpay4[i]=e102144[i]/(2.15*e98429[i]);

```

Appendix 2: Employment Variable Creation

```

if e102054[i] in (12,13) and e102144[i]>=0 then do;
/* condition based on the time unit to finish per othpay2. */
  if e10214aaa4[i]=3 and e10214aab4[i]>0 and e10214b4[i]>=0 and e98429[i]>0 then
    othpay4[i]=(e102144[i]/e10214aab4[i])*e10214b4[i]/e98429[i];
  if e10214aaa4[i]=4 and e10214aab4[i]>0 and e98429[i]>0 then othpay4[i]=(e102144[i]/e10214aab4[i])/e98429[i];
  if e10214aaa4[i]=5 and e10214aab4[i]>0 and e98429[i]>0 then
    othpay4[i]=(e102144[i]/e10214aab4[i])/(e98429[i]*4.3);
/* missing value */
  if e10214aaa4[i] in (3,4,5) and -4<e98429[i]<=0 then othpay4[i]=-3;
end;
if e102054[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e98429[i]=0 then othpay4[i]=-3;
if e102054[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e98429[i]<0 then
  othpay4[i]=e98429[i];
end;

/* non-hourly wage at the beginning, without overtime*/
do i=1 to 9;
if e98402[i]>0 and e102054[i]=2 and e10214b4[i]>0 and e102144[i]>=0 then
  othpay4[i]=(e102144[i]*e10214b4[i])/e98402[i];
if e98402[i]>0 and e102054[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e102144[i]>=0
  then othpay4[i]=e102144[i]/e98402[i];
if e98402[i]>0 and e102054[i]=4 and e102144[i]>=0 then othpay4[i]=e102144[i]/(2*e98402[i]);
if e98402[i]>0 and e102054[i]=5 and e102144[i]>=0 then othpay4[i]=e102144[i]/(4.3*e98402[i]);
if e98402[i]>0 and e102054[i]=6 and e102144[i]>=0 and e99500[i]>0 then
  othpay4[i]=e102144[i]/(e99500[i]*e98402[i]);
if e98402[i]>0 and e102054[i]=8 and e102144[i]>=0 then othpay4[i]=e102144[i]/(2.15*e98402[i]);
if e102054[i] in (7,9,12,13,15,16,17,14,18,19,21,22,23,24,25,26,28) then othpf4[i]=othpf4[i]+1;
  if e102054[i] in (12,13) and e102144[i]>=0 then do;
/* condition based on the time unit to finish per othpay2. */
  if e10214aaa4[i]=3 and e10214aab4[i]>0 and e10214b4[i]>=0 and e98402[i]>0 then
    othpay4[i]=(e102144[i]/e10214aab4[i])*e10214b4[i]/e98402[i];
  if e10214aaa4[i]=4 and e10214aab4[i]>0 and e98402[i]>0 then othpay4[i]=(e102144[i]/e10214aab4[i])/e98402[i];
  if e10214aaa4[i]=5 and e10214aab4[i]>0 and e98402[i]>0 then
    othpay4[i]=(e102144[i]/e10214aab4[i])/(e98402[i]*4.3);
/* missing value */
  if e10214aaa4[i] in (3,4,5) and -4<e98402[i]<=0 then othpay4[i]=-3;
end;
if e102054[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e98402[i]=0 then othpay4[i]=-3;
if e102054[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e98402[i]<0 then
  othpay4[i]=e98402[i];
if e102054[i]=6 and e99500[i]>0 then othpay4[i]=-3;
if e102054[i]=6 and -4<e99500[i]<0 then othpay4[i]=e99500[i];
end;

/* if incentive pay is corrected in the later part */
do i=1 to 9;
if e102258[i]=1 then do;
/*report hourly wage at the beginning*/
if e100237[i]=1 and e100253[i]>=0 then othpay4[i]=e100253[i];
if e100237[i]=2 and e100253[i]>=0 and e100253e[i]>0 and e88000[i]>0 then
  othpay4[i]=e100253[i]*e100253e[i]/e88000[i];
if e100237[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e100253[i]>=0 and e88000[i]>0
  then othpay4[i]=e100253[i]/e88000[i];
if e100237[i]=4 and e100253[i]>=0 and e88000[i]>0 then othpay4[i]=e100253[i]/(2*e88000[i]);
if e100237[i]=5 and e100253[i]>=0 and e88000[i]>0 then othpay4[i]=e100253[i]/(4.3*e88000[i]);

```

Appendix 2: Employment Variable Creation

```

if e100237[i]=6 and e100253[i]>=0 and e88000[i]>0 and e99500[i]>0 then
    othpay4[i]=e100253[i]/(e99500[i]*e88000[i]);
if e100237[i]=8 and e100253[i]>=0 and e88000[i]>0 then othpay4[i]=e100253[i]/(2.15*e88000[i]);
if e100237[i] in (9,14) then othpay4[i]=0;
if e100237[i] in (12,13) and e100253[i]>=0 then do;
    /* condition based on the time unit to finish per othpay4. */
    if e100253c[i]=1 and e100253cb[i]>0 then othpay4[i]=(e100253[i]/e100253cb[i])*60;
    if e100253c[i]=2 and e100253cb[i]>0 then othpay4[i]=e100253[i]/e100253cb[i];
    if e100253c[i]=3 and e100253cb[i]>0 and e100253e[i]>=0 and e88000[i]>0 then
        othpay4[i]=(e100253[i]/e100253cb[i])*e100253e[i]/e88000[i];
    if e100253c[i]=4 and e100253cb[i]>0 and e88000[i]>0 then othpay4[i]=(e100253[i]/e100253cb[i])/e88000[i];
    if e100253c[i]=5 and e100253cb[i]>0 and e88000[i]>0 then
        othpay4[i]=(e100253[i]/e100253cb[i])/(e88000[i]*4.3);
    /* missing value */
    if -4<e100253c[i]<0 or -4<e100253cb[i]<=0 then othpay4[i]=-3;
    if e100253c[i] in (3,4,5) and -4<e88000[i]<=0 then othpay4[i]=-3;
    if e100253c[i]=3 and e100253e[i]<0 then othpay4[i]=-3;
end;
if e100237[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e100253[i]<0 then
    othpay4[i]=e100253[i];
if e100237[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e88000[i]=0 then othpay4[i]=-3;
if e100237[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e88000[i]<0 then
    othpay4[i]=e88000[i];
if e100237[i]=2 and e100253e[i]<0 then othpay4[i]=-3;
if e100237[i]=2 and -4<e100253e[i]<0 then othpay4[i]=e100253e[i];

/* non-hourly wage at the beginning, with overtime*/
if e100237[i]=2 and e100253[i]>=0 and e100253e[i]>0 and e98429[i]>0 then
    othpay4[i]=e100253[i]*e100253e[i]/e98429[i];
if e100237[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e100253[i]>=0 and e98429[i]>0
    then othpay4[i]=e100253[i]/e98429[i];
if e100237[i]=4 and e100253[i]>=0 and e98429[i]>0 then othpay4[i]=e100253[i]/(2*e98429[i]);
if e100237[i]=5 and e100253[i]>=0 and e98429[i]>0 then othpay4[i]=e100253[i]/(4.3*e98429[i]);
if e100237[i]=6 and e100253[i]>=0 and e98429[i]>0 and e99500[i]>0 then
    othpay4[i]=e100253[i]/(e99500[i]*e98429[i]);
if e100237[i]=8 and e100253[i]>=0 and e98429[i]>0 then othpay4[i]=e100253[i]/(2.15*e98429[i]);
if e100237[i] in (12,13) and e100253[i]>=0 then do;
    /* condition based on the time unit to finish per othpay4. */
    if e100253c[i]=3 and e100253cb[i]>0 and e100253e[i]>=0 and e98429[i]>0 then
        othpay4[i]=(e100253[i]/e100253cb[i])*e100253e[i]/e98429[i];
    if e100253c[i]=4 and e100253cb[i]>0 and e98429[i]>0 then othpay4[i]=(e100253[i]/e100253cb[i])/e98429[i];
    if e100253c[i]=5 and e100253cb[i]>0 and e98429[i]>0 then
        othpay4[i]=(e100253[i]/e100253cb[i])/(e98429[i]*4.3);
    /* missing value */
    if e100253c[i] in (3,4,5) and -4<e98429[i]<=0 then othpay4[i]=-3;
end;
if e100237[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e98429[i]=0 then othpay4[i]=-3;
if e100237[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e98429[i]<0 then
    othpay4[i]=e98429[i];

/* non-hourly wage at the beginning, without overtime*/
if e100237[i]=2 and e100253[i]>=0 and e100253e[i]>0 and e98402[i]>0 then
    othpay4[i]=e100253[i]*e100253e[i]/e98402[i];
if e100237[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e100253[i]>=0 and e98402[i]>0
    then othpay4[i]=e100253[i]/e98402[i];
if e100237[i]=4 and e100253[i]>=0 and e98402[i]>0 then othpay4[i]=e100253[i]/(2*e98402[i]);
if e100237[i]=5 and e100253[i]>=0 and e98402[i]>0 then othpay4[i]=e100253[i]/(4.3*e98402[i]);

```

```

if e100237[i]=6 and e100253[i]>=0 and e98402[i]>0 and e99500[i]>0 then
    othpay4[i]=e100253[i]/(e99500[i]*e98402[i]);
if e100237[i]=8 and e100253[i]>=0 and e98402[i]>0 then othpay4[i]=e100253[i]/(2.15*e98402[i]);
if e100237[i] in (12,13) and e100253[i]>=0 then do;
    /* condition based on the time unit to finish per othpay4. */
    if e100253c[i]=3 and e100253cb[i]>0 and e100253e[i]>=0 and e98402[i]>0 then
        othpay4[i]=(e100253[i]/e100253cb[i])*e100253e[i]/e98402[i];
    if e100253c[i]=4 and e100253cb[i]>0 and e98402[i]>0 then othpay4[i]=(e100253[i]/e100253cb[i])/e98402[i];
    if e100253c[i]=5 and e100253cb[i]>0 and e98402[i]>0 then
        othpay4[i]=(e100253[i]/e100253cb[i])/(e98402[i]*4.3);
    /* missing value */
    if e100253c[i] in (3,4,5) and -4<e98402[i]<=0 then othpay4[i]=-3;
end;
if e100237[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e98402[i]=0 then othpay4[i]=-3;
if e100237[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e98402[i]<0 then
    othpay4[i]=e98402[i];
if e100237[i]=6 and e99500[i] le 0 then othpay4[i]=-3;
if e100237[i]=6 and -4<e99500[i]<0 then othpay4[i]=e99500[i];
end;
end;

***** for others *****

/*report hourly wage at the beginning*/
do i=1 to 9;
if e102055[i]=1 and e102145[i]>=0 then othpay5[i]=e102145[i];
if e88000[i]>0 and e102055[i]=2 and e10214b5[i]>0 and e102145[i]>=0 then
    othpay5[i]=(e102145[i]*e10214b5[i])/e88000[i];
if e88000[i]>0 and e102055[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e102145[i]>=0
    then othpay5[i]=e102145[i]/e88000[i];
if e88000[i]>0 and e102055[i]=4 and e102145[i]>=0 then othpay5[i]=e102145[i]/(2*e88000[i]);
if e88000[i]>0 and e102055[i]=5 and e102145[i]>=0 then othpay5[i]=e102145[i]/(4.3*e88000[i]);
if e88000[i]>0 and e102055[i]=6 and e102145[i]>=0 and e99500[i]>0 then
    othpay5[i]=e102145[i]/(e99500[i]*e88000[i]);
if e88000[i]>0 and e102055[i]=8 and e102145[i]>=0 then othpay5[i]=e102145[i]/(2.15*e88000[i]);
if e102055[i] in (9,14) then othpay5[i]=0;
if e102055[i] in (12,13) and e102145[i]>=0 then do;
    /* condition based on the time unit to finish per othpay2. */
    if e10214aaa5[i]=1 and e10214aab5[i]>0 then othpay5[i]=(e102145[i]/e10214aab5[i])*60;
    if e10214aaa5[i]=2 and e10214aab5[i]>0 then othpay5[i]=e102145[i]/e10214aab5[i];
    if e10214aaa5[i]=3 and e10214aab5[i]>0 and e10214b5[i]>=0 and e88000[i]>0 then
        othpay5[i]=(e102145[i]/e10214aab5[i])*e10214b5[i]/e88000[i];
    if e10214aaa5[i]=4 and e10214aab5[i]>0 and e88000[i]>0 then othpay5[i]=(e102145[i]/e10214aab5[i])/e88000[i];
    if e10214aaa5[i]=5 and e10214aab5[i]>0 and e88000[i]>0 then
        othpay5[i]=(e102145[i]/e10214aab5[i])/(e88000[i]*4.3);
    /* missing value */
    if -4<e10214aaa5[i]<0 or -4<e10214aab5[i]<=0 then othpay5[i]=-3;
    if e10214aaa5[i] in (3,4,5) and -4<e88000[i]<=0 then othpay5[i]=-3;
    if e10214aaa5[i]=3 and e10214b5[i]<0 then othpay5[i]=-3;
end;
if e102055[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e102145[i]<0 then
    othpay5[i]=e102145[i];
if e102055[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e88000[i]=0 then othpay5[i]=-3;
if e102055[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e88000[i]<0 then
    othpay5[i]=e88000[i];
if e102055[i]=2 and e10214b5[i] le 0 then othpay5[i]=-3;
if e102055[i]=2 and -4<e10214b5[i]<0 then othpay5[i]=e10214b5[i];

```

```

end;

/* non-hourly wage at the beginning, with overtime*/
do i=1 to 9;
if e98429[i]>0 and e102055[i]=2 and e10214b5[i]>0 and e102145[i]>=0 then
    othpay5[i]=(e102145[i]*e10214b5[i])/e98429[i];
if e98429[i]>0 and e102055[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e102145[i]>=0
    then othpay5[i]=e102145[i]/e98429[i];
if e98429[i]>0 and e102055[i]=4 and e102145[i]>=0 then othpay5[i]=e102145[i]/(2*e98429[i]);
if e98429[i]>0 and e102055[i]=5 and e102145[i]>=0 then othpay5[i]=e102145[i]/(4.3*e98429[i]);
if e98429[i]>0 and e102055[i]=6 and e102145[i]>=0 and e99500[i]>0 then
    othpay5[i]=e102145[i]/(e99500[i]*e98429[i]);
if e98429[i]>0 and e102055[i]=8 and e102145[i]>=0 then othpay5[i]=e102145[i]/(2.15*e98429[i]);
if e102055[i] in (12,13) and e102145[i]>=0 then do;
    /* condition based on the time unit to finish per othpay2.*/
    if e10214aaa5[i]=3 and e10214aab5[i]>0 and e10214b5[i]>=0 and e98429[i]>0 then
        othpay5[i]=(e102145[i]/e10214aab5[i])*e10214b5[i]/e98429[i];
    if e10214aaa5[i]=4 and e10214aab5[i]>0 and e98429[i]>0 then othpay5[i]=(e102145[i]/e10214aab5[i])/e98429[i];
    if e10214aaa5[i]=5 and e10214aab5[i]>0 and e98429[i]>0 then
        othpay5[i]=(e102145[i]/e10214aab5[i])/(e98429[i]*4.3);
    /* missing value */
    if e10214aaa5[i] in (3,4,5) and -4<e98429[i]<=0 then othpay5[i]=-3;
end;
if e102055[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999, -1, -2) and e98429[i]=0 then othpay5[i]=-3;
if e102055[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999, -1, -2) and -4<e98429[i]<0 then
    othpay5[i]=e98429[i];

end;

/* non-hourly wage at the beginning, without overtime*/
do i=1 to 9;
if e98402[i]>0 and e102055[i]=1 and e102145[i]>=0 then othpay5[i]=e102145[i];
if e98402[i]>0 and e102055[i]=2 and e10214b5[i]>0 and e102145[i]>=0 then
    othpay5[i]=(e102145[i]*e10214b5[i])/e98402[i];
if e98402[i]>0 and e102055[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e102145[i]>=0
    then othpay5[i]=e102145[i]/e98402[i];
if e98402[i]>0 and e102055[i]=4 and e102145[i]>=0 then othpay5[i]=e102145[i]/(2*e98402[i]);
if e98402[i]>0 and e102055[i]=5 and e102145[i]>=0 then othpay5[i]=e102145[i]/(4.3*e98402[i]);
if e98402[i]>0 and e102055[i]=6 and e102145[i]>=0 and e99500[i]>0 then
    othpay5[i]=e102145[i]/(e99500[i]*e98402[i]);
if e98402[i]>0 and e102055[i]=8 and e102145[i]>=0 then othpay5[i]=e102145[i]/(2.15*e98402[i]);
if e102055[i] in (7,9,12,13,15,16,17,14,18,19,21,22,23,24,25,26,28) then othpf5[i]=othpf5[i]+1;
if e102055[i] in (9,14) then othpay5[i]=0;
if e102055[i] in (12,13) and e102145[i]>=0 then do;
    /* condition based on the time unit to finish per othpay2.*/
    if e10214aaa5[i]=3 and e10214aab5[i]>0 and e10214b5[i]>=0 and e98402[i]>0 then
        othpay5[i]=(e102145[i]/e10214aab5[i])*e10214b5[i]/e98402[i];
    if e10214aaa5[i]=4 and e10214aab5[i]>0 and e98402[i]>0 then othpay5[i]=(e102145[i]/e10214aab5[i])/e98402[i];
    if e10214aaa5[i]=5 and e10214aab5[i]>0 and e98402[i]>0 then
        othpay5[i]=(e102145[i]/e10214aab5[i])/(e98402[i]*4.3);
    /* missing value */
    if e10214aaa5[i] in (3,4,5) and -4<e98402[i]<=0 then othpay5[i]=-3;
end;
if e102055[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999, -1, -2) and e98402[i]=0 then othpay5[i]=-3;
if e102055[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999, -1, -2) and -4<e98402[i]<0 then
    othpay5[i]=e98402[i];
if e102055[i]=6 and e99500[i] le 0 then othpay5[i]=-3;

```

```

if e102055[i]=6 and -4<e99500[i]<0 then othpay5[i]=e99500[i];
end;

/* if other compensation is corrected in the later part */
do i=1 to 9;
if e102259[i]=1 then do;

/*report hourly wage at the beginning*/
if e100239[i]=1 and e100254[i]>=0 then othpay5[i]=e100254[i];
if e100239[i]=2 and e100254[i]>=0 and e100254e[i]>0 and e88000[i]>0 then
    othpay5[i]=e100254[i]*e100254e[i]/e88000[i];
if e100239[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e100254[i]>=0 and e88000[i]>0
    then othpay5[i]=e100254[i]/e88000[i];
if e100239[i]=4 and e100254[i]>=0 and e88000[i]>0 then othpay5[i]=e100254[i]/(2*e88000[i]);
if e100239[i]=5 and e100254[i]>=0 and e88000[i]>0 then othpay5[i]=e100254[i]/(4.3*e88000[i]);
if e100239[i]=6 and e100254[i]>=0 and e88000[i]>0 and e99500[i]>0 then
    othpay5[i]=e100254[i]/(e99500[i]*e88000[i]);
if e100239[i]=8 and e100254[i]>=0 and e88000[i]>0 then othpay5[i]=e100254[i]/(2.15*e88000[i]);
if e100239[i] in (9,14) then othpay5[i]=0;
if e100239[i] in (12,13) and e100254[i]>=0 then do;
/* condition based on the time unit to finish per othpay5. */
if e100254c[i]=1 and e100254cb[i]>0 then othpay5[i]=(e100254[i]/e100254cb[i])*60;
if e100254c[i]=2 and e100254cb[i]>0 then othpay5[i]=e100254[i]/e100254cb[i];
if e100254c[i]=3 and e100254cb[i]>0 and e100254e[i]>0 and e88000[i]>0 then
    othpay5[i]=(e100254[i]/e100254cb[i])*e100254e[i]/e88000[i];
if e100254c[i]=4 and e100254cb[i]>0 and e88000[i]>0 then othpay5[i]=(e100254[i]/e100254cb[i])/e88000[i];
if e100254c[i]=5 and e100254cb[i]>0 and e88000[i]>0 then
    othpay5[i]=(e100254[i]/e100254cb[i])/(e88000[i]*4.3);
/* missing value */
if -4<e100254c[i]<0 or -4<e100254cb[i]<=0 then othpay5[i]=-3;
if e100254c[i] in (3,4,5) and -4<e88000[i]<=0 then othpay5[i]=-3;
if e100254c[i]=3 and e100254e[i]<0 then othpay5[i]=-3;
end;
if e100239[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e100254[i]<0 then
    othpay5[i]=e100254[i];
if e100239[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e88000[i]=0 then othpay5[i]=-3;
if e100239[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e88000[i]<0 then
    othpay5[i]=e88000[i];
if e100239[i]=2 and e100254e[i] le 0 then othpay5[i]=-3;
if e100239[i]=2 and -4<e100254e[i]<0 then othpay5[i]=e100254e[i];

/* non-hourly wage at the beginning, with overtime*/
if e100239[i]=2 and e100254[i]>=0 and e100254e[i]>0 and e98429[i]>0 then
    othpay5[i]=e100254[i]*e100254e[i]/e98429[i];
if e100239[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e100254[i]>=0 and e98429[i]>0
    then othpay5[i]=e100254[i]/e98429[i];
if e100239[i]=4 and e100254[i]>=0 and e98429[i]>0 then othpay5[i]=e100254[i]/(2*e98429[i]);
if e100239[i]=5 and e100254[i]>=0 and e98429[i]>0 then othpay5[i]=e100254[i]/(4.3*e98429[i]);
if e100239[i]=6 and e100254[i]>=0 and e98429[i]>0 and e99500[i]>0 then
    othpay5[i]=e100254[i]/(e99500[i]*e98429[i]);
if e100239[i]=8 and e100254[i]>=0 and e98429[i]>0 then othpay5[i]=e100254[i]/(2.15*e98429[i]);
if e100239[i] in (12,13) and e100254[i]>=0 then do;
/* condition based on the time unit to finish per othpay5. */
if e100254c[i]=3 and e100254cb[i]>0 and e100254e[i]>0 and e98429[i]>0 then
    othpay5[i]=(e100254[i]/e100254cb[i])*e100254e[i]/e98429[i];
if e100254c[i]=4 and e100254cb[i]>0 and e98429[i]>0 then othpay5[i]=(e100254[i]/e100254cb[i])/e98429[i];

```

Appendix 2: Employment Variable Creation

```

if e100254c[i]=5 and e100254cb[i]>0 and e98429[i]>0 then
    othpay5[i]=(e100254[i]/e100254cb[i])/(e98429[i]*4.3);
/* missing value */
if e100254c[i] in (3,4,5) and -4<e98429[i]<=0 then othpay5[i]=-3;
end;
if e100239[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e98429[i]=0 then othpay5[i]=-3;
if e100239[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e98429[i]<0 then
    othpay5[i]=e98429[i];

/* non-hourly wage at the beginning, without overtime*/
if e100239[i]=2 and e100254[i]>=0 and e100254e[i]>0 and e98402[i]>0 then
    othpay5[i]=e100254[i]*e100254e[i]/e98402[i];
if e100239[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e100254[i]>=0 and e98402[i]>0
    then othpay5[i]=e100254[i]/e98402[i];
if e100239[i]=4 and e100254[i]>=0 and e98402[i]>0 then othpay5[i]=e100254[i]/(2*e98402[i]);
if e100239[i]=5 and e100254[i]>=0 and e98402[i]>0 then othpay5[i]=e100254[i]/(4.3*e98402[i]);
if e100239[i]=6 and e100254[i]>=0 and e98402[i]>0 and e99500[i]>0 then
    othpay5[i]=e100254[i]/(e99500[i]*e98402[i]);
if e100239[i]=8 and e100254[i]>=0 and e98402[i]>0 then othpay5[i]=e100254[i]/(2.15*e98402[i]);
if e100239[i] in (12,13) and e100254[i]>=0 then do;
    /* condition based on the time unit to finish per othpay5. */
    if e100254c[i]=3 and e100254cb[i]>0 and e100254e[i]>=0 and e98402[i]>0 then
        othpay5[i]=(e100254[i]/e100254cb[i])*e100254e[i]/e98402[i];
    if e100254c[i]=4 and e100254cb[i]>0 and e98402[i]>0 then othpay5[i]=(e100254[i]/e100254cb[i])/e98402[i];
    if e100254c[i]=5 and e100254cb[i]>0 and e98402[i]>0 then
        othpay5[i]=(e100254[i]/e100254cb[i])/(e98402[i]*4.3);
/* missing value */
    if e100254c[i] in (3,4,5) and -4<e98402[i]<=0 then othpay5[i]=-3;
end;
if e100239[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e98402[i]=0 then othpay5[i]=-3;
if e100239[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e98402[i]<0 then
    othpay5[i]=e98402[i];
if e100239[i]=6 and e99500[i] le 0 then othpay5[i]=-3;
if e100239[i]=6 and -4<e99500[i]<0 then othpay5[i]=e99500[i];

end;
end;

***** overall start hourly compensation *****/
array hrcomp hrcomp01 hrcomp02 hrcomp03 hrcomp04 hrcomp05 hrcomp06 hrcomp07 hrcomp08 hrcomp09;
do i=1 to 9;
    hrcomp[i]=0;
    if hrwg[i] ge 0 then hrcomp[i]=hrcomp[i]+hrwg[i];
    if ot[i] ge 0 then hrcomp[i]=hrcomp[i]+ot[i];
    if othpay1[i] ge 0 then hrcomp[i]=hrcomp[i]+othpay1[i];
    if othpay2[i] ge 0 then hrcomp[i]=hrcomp[i]+othpay2[i];
    if othpay3[i] ge 0 then hrcomp[i]=hrcomp[i]+othpay3[i];
    if othpay4[i] ge 0 then hrcomp[i]=hrcomp[i]+othpay4[i];
    if othpay5[i] ge 0 then hrcomp[i]=hrcomp[i]+othpay5[i];
    if -4<hrwg[i]<0 or -4<ot[i]<0 or -4<othpay1[i]<0 or -4<othpay2[i]<0 or -4<othpay3[i]<0 or -4<othpay4[i]<0 or -
        4<othpay5[i]<0 then hrcomp[i]=-3;
    if hrwg[i]=-4 then hrcomp[i]=-4;
end;

***** end wages for youth *****/

```

Appendix 2: Employment Variable Creation

*****first, find the regular working hours for the people who have no overtime at the beginning;
array e3800ff e3800ff1-e3800ff9; /*which is the regular hours.*/

```
do i=1 to 9;
  if e38002[i] in (0,-4) or e3800f[i] in (0,-4) then e3800ff[i]=e3800f[i];
  if -4<e38002[i]<0 or -4<e3800f[i]<0 then e3800ff[i]=-3;
  if e38002[i]>0 and e3800f[i]>0 then e3800ff[i]=e3800f[i]-e38002[i];
end;
```

/*set up the end hourly wage for youths who report an hourly wage*/

```
do i=1 to 9;
```

/* end hourly wage for workers */

```
if (e37901b[i]=1 or e59900[i]=1) then do;
```

/* without overtime at the beginning*/

```
  if (e38013[i]=1 and e38014[i]=1 ) then do;
    if e38023[i]>=0 then hrwage[i]=e38023[i];
    /*missing value*/
    if -4<e38023[i]<0 then hrwage[i]=e38023[i];
  end;
```

/* with overtime at the beginning */

```
  if (e38106[i]=1 and e38107[i]=1) then do;
    if e38116[i]>=0 then hrwage[i]=e38116[i];
    /*missing value*/
    if -4<e38106[i]<0 then hrwage[i]=e38106[i];
  end;
end;
```

```
end;
```

/*set up the end hourly wage for youths who report their wage in daily units.*/

```
do i=1 to 9;
```

/* end hourly wage for workers */

```
if (e37901b[i]=1 or e59900[i]=1) then do;
```

/*without overtime at the beginning*/

```
  if (e38013[i]=1 and e38014[i]=2) then do;
    if e38023[i]>=0 and e3800b[i]=1 and e38027[i]>0 and e34402[i]>0 then daily[i]=e38023[i]*e38027[i]/e34402[i];
    if e38023[i]>=0 and e3800ff[i]>0 and e38027[i]>0 then daily[i]=e38023[i]*e38027[i]/e3800ff[i];
    /*missing value*/
    if e38023[i]>=0 and -4<e38027[i]<0 then daily[i]=e38027[i];
    if e38023[i]>=0 and e3800b[i]=1 and -4<e34402[i]<0 then daily[i]=e34402[i];
    if e38023[i]>=0 and -4<e3800ff[i]<0 then daily[i]=e3800ff[i];
    if e38023[i]>=0 and (e34402[i]=0 or e3800ff[i]=0) then daily[i]=-3;
  end;
```

/*with overtime at the beginning*/

```
  if (e38106[i]=1 and e38107[i]=2) then do;
    if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and e38116b[i]>0 and e34428[i]>0 then
      daily[i]=e38116b[i]*e38116[i]/e34428[i];
    if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and e38103[i]>0 and e38116b[i]>0 then
      daily[i]=e38116b[i]*e38116[i]/e38103[i];
    /*missing value*/
  end;
```

Appendix 2: Employment Variable Creation

```
if e38116[i]>=0 and -4<e38116b[i]<0 then daily[i]=e38116b[i];
if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and -4<e34428[i]<0 then daily[i]=e34428[i];
if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and -4<e38103[i]<0 then daily[i]=e38103[i];
if e38116[i]>=0 and (e34428[i]=0 or e38103[i]=0) then daily[i]=-3;
end;
end;

end;

/*set up the end hourly wage for youths who report their wage in weekly units.*/

do i=1 to 9;

/* end hourly wage for workers */

if (e37901b[i]=1 or e59900[i]=1) then do;

/*without overtime at the beginning */
if e38013[i]=1 and e38014[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) then do;
if e38023[i]>=0 and e3800b[i]=1 and e34402[i]>0 then weekly[i]=e38023[i]/e34402[i];
if e38023[i]>=0 and e3800ff[i]>0 then weekly[i]=e38023[i]/e3800ff[i];
/*missing value*/
if e38023[i]>=0 and e3800b[i]=1 and -4<e34402[i]<0 then weekly[i]=e34402[i];
if e38023[i]>=0 and -4<e3800ff[i]<0 then weekly[i]=e3800ff[i];
if e38023[i]>=0 and (e34402[i]=0 or e3800ff[i]=0) then weekly[i]=-3;
end;

/* with overtime at the beginning */
if e38106[i]=1 and e38107[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) then do;
if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and e34428[i]>0 then weekly[i]=e38116[i]/e34428[i];
if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and e38103[i]>0 then weekly[i]=e38116[i]/e38103[i];
/*missing value*/
if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and -4<e34428[i]<0 then weekly[i]=e34428[i];
if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and -4<e38103[i]<0 then weekly[i]=e38103[i];
if e38116[i]>=0 and (e34428[i]=0 or e38103[i]=0) then weekly[i]=-3;
end;
end;
end;

/*set up the end hourly wage for youths who report their wage in biwkly units.*/

do i=1 to 9;

/* end hourly wage for workers */

if (e37901b[i]=1 or e59900[i]=1) then do;

/* without overtime at the beginning */
if (e38013[i]=1 and e38014[i]=4) then do;
if e38023[i]>=0 and e3800b[i]=1 and e34402[i]>0 then biwkly[i]=e38023[i]/(2*e34402[i]);
if e38023[i]>=0 and e3800ff[i]>0 then biwkly[i]=e38023[i]/(2*e3800ff[i]);
/*missing value*/
if e38023[i]>=0 and e3800b[i]=1 and -4<e34402[i]<0 then biwkly[i]=e34402[i];
if e38023[i]>=0 and -4<e3800ff[i]<0 then biwkly[i]=e3800ff[i];
if e38023[i]>=0 and (e34402[i]=0 or e3800ff[i]=0) then biwkly[i]=-3;
end;

/* with overtime at the beginning*/
if (e38106[i]=1 and e38107[i]=4) then do;
```

Appendix 2: Employment Variable Creation

```
if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and e34428[i]>0 then biwkly[i]=e38116[i]/(2*e34428[i]);
if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and e38103[i]>0 then biwkly[i]=e38116[i]/(2*e38103[i]);
/*missing value*/
if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and -4<e34428[i]<0 then biwkly[i]=e34428[i];
if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and -4<e38103[i]<0 then biwkly[i]=e38103[i];
if e38116[i]>=0 and (e34428[i]=0 or e38103[i]=0) then biwkly[i]=-3;
end;
end;

end;

/*set up the end hourly wage for youths who report their wage in month units.*/

do i=1 to 9;

/* end hourly wage for workers */

if (e37901b[i]=1 or e59900[i]=1) then do;

/* without overtime at the beginning */
if (e38013[i]=1 and e38014[i]=5) then do;
if e38023[i]>=0 and e3800b[i]=1 and e34402[i]>0 then month[i]=e38023[i]/(4.3*e34402[i]);
if e38023[i]>=0 and e3800ff[i]>0 then month[i]=e38023[i]/(4.3*e3800ff[i]);
/*missing value*/
if e38023[i]>=0 and e3800b[i]=1 and -4<e34402[i]<0 then month[i]=e34402[i];
if e38023[i]>=0 and -4<e3800ff[i]<0 then month[i]=e3800ff[i];
if e38023[i]>=0 and (e34402[i]=0 or e3800ff[i]=0) then month[i]=-3;
end;
/* with overtime at the beginning */
if (e38106[i]=1 and e38107[i]=5) then do;
if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and e34428[i]>0 then month[i]=e38116[i]/(4.3*e34428[i]);
if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and e38103[i]>0 then month[i]=e38116[i]/(4.3*e38103[i]);
/*missing value*/
if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and -4<e34428[i]<0 then month[i]=e34428[i];
if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and -4<e38103[i]<0 then month[i]=e38103[i];
if e38116[i]>=0 and (e34428[i]=0 or e38103[i]=0) then month[i]=-3;
end;
end;

end;

/*set up the end hourly wage for youths who report their wage in annual units.*/

do i=1 to 9;

/* end hourly wage for workers */

if (e37901b[i]=1 or e59900[i]=1) then do;

/* without overtime at the beginning */
if (e38013[i]=1 and e38014[i]=6) then do;
if e38023[i]>=0 and e3800b[i]=1 and e34402[i]>0 and e35600[i]>0 then annual[i]=e38023[i]/(e35600[i]*e34402[i]);
if e38023[i]>=0 and e3800ff[i]>0 and e35600[i]>0 then annual[i]=e38023[i]/(e35600[i]*e3800ff[i]);
/*missing value*/
if e38023[i]>=0 and e3800b[i]=1 and -4<e34402[i]<0 then annual[i]=e34402[i];
if e38023[i]>=0 and -4<e3800ff[i]<0 then annual[i]=e3800ff[i];
if e38023[i]>=0 and (e34402[i]=0 or e3800ff[i]=0) then annual[i]=-3;
```

Appendix 2: Employment Variable Creation

```
if e35600[i] le 0 then annual[i]=-3;
if -4<e35600[i]<0 then annual[i]=e35600[i];
end;
/* with overtime at the beginning */
if (e38106[i]=1 and e38107[i]=6) then do;
  if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and e34428[i]>0 and e35600[i]>0 then
    annual[i]=e38116[i]/(e35600[i]*e34428[i]);
  if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and e38103[i]>0 and e35600[i]>0 then
    annual[i]=e38116[i]/(e35600[i]*e38103[i]);
  /*missing value*/
  if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and -4<e34428[i]<0 then annual[i]=e34428[i];
  if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and -4<e38103[i]<0 then annual[i]=e38103[i];
  if e38116[i]>=0 and (e34428[i]=0 or e38103[i]=0) then annual[i]=-3;
  if e35600[i] le 0 then annual[i]=-3;
  if -4<e35600[i]<0 then annual[i]=e35600[i];
end;
end;

/*set up the end hourly wage for youths who report their wage in semim units.*/
do i=1 to 9;

/* end hourly wage for workers */

if (e37901b[i]=1 or e59900[i]=1) then do;

/* without overtime at the beginning*/
if (e38013[i]=1 and e38014[i]=8) then do;
  if e38023[i]>=0 and e3800b[i]=1 and e34402[i]>0 then semim[i]=e38023[i]/(2.15*e34402[i]);
  if e38023[i]>=0 and e3800ff[i]>0 then semim[i]=e38023[i]/(2.15*e3800ff[i]);
  /*missing value*/
  if e38023[i]>=0 and e3800b[i]=1 and -4<e34402[i]<0 then semim[i]=e34402[i];
  if e38023[i]>=0 and -4<e3800ff[i]<0 then semim[i]=e3800ff[i];
  if e38023[i]>=0 and (e34402[i]=0 or e3800ff[i]=0) then semim[i]=-3;
end;
/* with overtime at the beginning */
if (e38106[i]=1 and e38107[i]=8) then do;
  if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and e34428[i]>0 then
    semim[i]=e38116[i]/(2.15*e34428[i]);
  if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and e38103[i]>0 then semim[i]=e38116[i]/(2.15*e38103[i]);
  /*missing value*/
  if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) and -4<e34428[i]<0 then semim[i]=e34428[i];
  if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) and -4<e38103[i]<0 then semim[i]=e38103[i];
  if e38116[i]>=0 and (e34428[i]=0 or e38103[i]=0) then semim[i]=-3;
end;
end;

/*set up the end hourly wage for youths who report their wage in per item.*/
do i=1 to 9;

/* end hourly wage for workers */
```

```

if (e37901b[i]=1 or e59900[i]=1) then do;

/* without overtime at the beginning */
if (e38013[i]=1 and e38014[i] in (12,13)) then do;
if e38023[i]>=0 then endhr1[i]=e3800ff[i];
if e38023[i]>=0 and e3800b[i]=1 then endhr1[i]=e34402[i];
/* condition based on the time unit to finish per item.*/
if e38024b[i]=1 and e38024c[i]>0 then item[i]=(e38023[i]/e38024c[i])*60;
if e38024b[i]=2 and e38024c[i]>0 then item[i]=e38023[i]/e38024c[i];
if e38024b[i]=3 and e38024c[i]>0 and e38027[i]>=0 and endhr1[i]>0 then
    item[i]=(e38023[i]/e38024c[i])*e38027[i]/endhr1[i];
if e38024b[i]=4 and e38024c[i]>0 and endhr1[i]>0 then item[i]=(e38023[i]/e38024c[i])/endhr1[i];
if e38024b[i]=5 and e38024c[i]>0 and endhr1[i]>0 then item[i]=(e38023[i]/e38024c[i])/(endhr1[i]*4.3);
/* missing value */
if -4<e38024b[i]<0 or -4<e38024c[i]<=0 then item[i]=-3;
if e38024b[i] in (3,4,5) and -4<endhr1[i]<=0 then item[i]=-3;
if e38024b[i]=3 and e38027[i]<0 then item[i]=-3;
end;
/* with overtime at the beginning */
if (e38106[i]=1 and e38107[i] in (12,13)) then do;
if e38116[i]>=0 and (e38102[i] ne 1 and e38102[i] ne 3) then endhr2[i]=e34428[i];
if e38116[i]>=0 and (e38102[i]=1 or e38102[i]=3) then endhr2[i]=e38103[i];
/* condition based on the time unit to finish per item.*/
if e38116aaa[i]=1 and e38116aab[i]>0 then item[i]=(e38116[i]/e38116aab[i])*60;
if e38116aaa[i]=2 and e38116aab[i]>0 then item[i]=e38116[i]/e38116aab[i];
if e38116aaa[i]=3 and e38116aab[i]>0 and e38116b[i]>=0 and endhr2[i]>0 then
    item[i]=(e38116[i]/e38116aab[i])*e38116b[i]/endhr2[i];
if e38116aaa[i]=4 and e38116aab[i]>0 and endhr2[i]>0 then item[i]=(e38116[i]/e38116aab[i])/endhr2[i];
if e38116aaa[i]=5 and e38116aab[i]>0 and endhr2[i]>0 then item[i]=(e38116[i]/e38116aab[i])/(endhr2[i]*4.3);
/* missing value */
if -4<e38116aaa[i]<0 or -4<e38116aab[i]<=0 then item[i]=-3;
if e38116aaa[i] in (3,4,5) and -4<endhr2[i]<=0 then item[i]=-3;
if e38116aaa[i]=3 and e38116b[i]<0 then item[i]=-3;
end;
end;
end;

***** create the hourly rate of pay combining the information from end wage *****/
do i=1 to 9;

if e37901b[i]=1 or e59900[i]=1 then do;

if annual[i] ge 0 then hrwg[i]=annual[i];
if month[i] ge 0 then hrwg[i]=month[i];
if biwkly[i] ge 0 then hrwg[i]=biwkly[i];
if weekly[i] ge 0 then hrwg[i]=weekly[i];
if daily[i] ge 0 then hrwg[i]=daily[i];
if hrwage[i] ge 0 then hrwg[i]=hrwage[i];
if semim[i] ge 0 then hrwg[i]=semim[i];
if item[i] ge 0 then hrwg[i]=item[i];
if hrwage[i] eq -1 or daily[i]=-1 or weekly[i] eq -1 or biwkly[i] eq -1 or month[i] eq -1 or
    annual[i] eq -1 or semim[i]=-1 or item[i]=-1 then hrwg[i]=-1;
if hrwage[i] eq -2 or daily[i]=-2 or weekly[i] eq -2 or biwkly[i] eq -2 or month[i] eq -2 or
    annual[i] eq -2 or semim[i]=-2 or item[i]=-2 then hrwg[i]=-2;
if hrwage[i] eq -3 or daily[i]=-3 or weekly[i] eq -3 or biwkly[i] eq -3 or month[i] eq -3 or
    annual[i] eq -3 or semim[i]=-3 or item[i]=-3 then hrwg[i]=-3;

```

Appendix 2: Employment Variable Creation

```
end;
end;

/*set up the hourly wage for youths who report their wage in other manners */

do i=1 to 9.;

/* end hourly wage for workers */

if (e37901b[i]=1 or e59900[i]=1) then do;

/* if job was not offering any comp at the end */
  if (e38013[i]=1 and e38014[i] in (9,14)) then hrwg[i]=0;
  if e38013[i]=1 and -4<e38023[i]<0 then hrwg[i]=e38023[i];
/* if job was offering a comp at the end */
  if (e38106[i]=1 and e38107[i] in (9,14)) then hrwg[i]=0;
  if e38106[i]=1 and -4<e38116[i]<0 then hrwg[i]=e38116[i];

end;
end;

/* report -1,-2 or -3 if the amount reported is -1, -2 or -3*/
do i=1 to 9;
  if e37901b[i]=1 or e59900[i]=1 then do;
    if -4<e38023[i]<0 then hrwg[i]=e38023[i];
    if -4<e38116[i]<0 then hrwg[i]=e38116[i];
  end;
end;

/* the end wage if job lasts for >=13 weeks and report the same amount but diff. hours*/

/* note: for this case, we only change hourly rate hrwg[i] without changing hrwage[i], daily[i], weeklw[i] biwkly[i],
month[i], annual[i] or semim[i] */

do i=1 to 9;
  if ((e37901b[i]=1 or e59900[i]=1) and e38013[i]=0 and e3800b[i]=0) and hrwg[i]>=0
    and e34402[i]>0 and e3800ff[i]>0 then hrwg[i]=hrwg[i]*e34402[i]/e3800ff[i];
  if ((e37901b[i]=1 or e59900[i]=1) and e38013[i]=0 and e3800b[i]=0) and e3800ff[i]=0 then hrwg[i]=-3;
  if ((e37901b[i]=1 or e59900[i]=1) and e38013[i]=0 and e3800b[i]=0) and -4<e3800ff[i]<0 then hrwg[i]=e3800ff[i];
  if ((e37901b[i]=1 or e59900[i]=1) and e38106[i]=0 and (e38102[i]=1 or e38102[i]=3))
    and hrwg[i]>=0 and e34428[i]>0 and e38103[i]>0 then hrwg[i]=hrwg[i]*e34428[i]/e38103[i];
  if ((e37901b[i]=1 or e59900[i]=1) and e38106[i]=0 and (e38102[i]=1 or e38102[i]=3))
    and hrwg[i]>=0 and e34402[i]>0 and e38103[i]>0 then hrwg[i]=hrwg[i]*e34402[i]/e38103[i];
  if ((e37901b[i]=1 or e59900[i]=1) and e38106[i]=0 and (e38102[i]=1 or e38102[i]=3))
    and e38103[i]=0 then hrwg[i]=-3;
  if ((e37901b[i]=1 or e59900[i]=1) and e38106[i]=0 and (e38102[i]=1 or e38102[i]=3))
    and -4<e38103[i]<0 then hrwg[i]=e38103[i];
end;

***** add the end compensation *****
***** amount paid of overtime *****

do i=1 to 9;

if (e37901b[i]=1 or e59900[i]=1) then do;
```

```

***** without compensation at the beginning *****

if e38001[i]=0 then ot[i]=0;

else do;
if e38003[i]=0 then multot=multot+1;
if e38002[i]>0 and e38003[i]=1 and e38012[i]>=0 then ot[i]=e38012[i];
if e38002[i]>0 and e38003[i]=2 and e38012[i]>=0 and e38012b[i]>0 then ot[i]=e38012[i]*e38012b[i]/e38002[i];
if e38002[i]>0 and e38003[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38012[i]>=0 then
    ot[i]=e38012[i]/e38002[i];
if e38002[i]>0 and e38003[i]=4 and e38012[i]>=0 then ot[i]=e38012[i]/(2*e38002[i]);
if e38002[i]>0 and e38003[i]=5 and e38012[i]>=0 then ot[i]=e38012[i]/(4.3*e38002[i]);
if e38002[i]>0 and e38003[i]=6 and e38012[i]>=0 then ot[i]=-3; /* since no weeks per year available. */
if e38002[i]>0 and e38003[i]=8 and e38012[i]>=0 then ot[i]=e38012[i]/(2.15*e38002[i]);
if e38003[i] in (9,14) then ot[i]=0;
if e38003[i] in (12,13) and e38012[i]>=0 then do;
/* condition based on the time unit to finish per ot. */
    if e38012aaa[i]=1 and e38012aab[i]>0 then ot[i]=(e38012[i]/e38012aab[i])*60;
    if e38012aaa[i]=2 and e38012aab[i]>0 then ot[i]=e38012[i]/e38012aab[i];
    if e38012aaa[i]=3 and e38012aab[i]>0 and e38012b[i]>0 and e38002[i]>0 then
        ot[i]=(e38012[i]/e38012aab[i])*e38012b[i]/e38002[i];
    if e38012aaa[i]=4 and e38012aab[i]>0 and e38002[i]>0 then ot[i]=(e38012[i]/e38012aab[i])/e38002[i];
    if e38012aaa[i]=5 and e38012aab[i]>0 and e38002[i]>0 then ot[i]=(e38012[i]/e38012aab[i])/(e38002[i]*4.3);
/* missing value */
    if -4<e38012aaa[i]<0 or -4<e38012aab[i]<=0 then ot[i]=-3;
    if e38012aaa[i] in (3,4,5) and -4<e38002[i]<=0 then ot[i]=-3;
    if e38012aaa[i]=3 and e38012b[i]<0 then ot[i]=-3;
end;

/*missing value*/
if -4<e38002[i]<0 then ot[i]=e38002[i];
if e38012[i]>=0 and e38002[i]=0 then ot[i]=-3;
if -4<e38012[i]<0 then ot[i]=e38012[i];
if e38003[i]=2 and e38012b[i]<=0 then ot[i]=-3;
if e38003[i]=2 and -4<e38012b[i]<0 then ot[i]=e38012b[i];
end;

***** with compensation at the beginning *****

if e38201[i]=1 then do;
if e38202[i]=0 then multot=multot+1;
if e38101[i]=1 then do; /*same no. of hours as at the beginning */

if e19200[i]=1 then do; /* report hourly rate of pay at the beginning*/

    if e24501[i]>0 and e38202[i]=1 and e38211[i]>=0 then ot[i]=e38211[i];
    if e24501[i]>0 and e38202[i]=2 and e38211b[i]>0 and e38211[i]>=0 then ot[i]=e38211[i]*e38211b[i]/e24501[i];
    if e24501[i]>0 and e38202[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38211[i]>=0
        then ot[i]=e38211[i]/e24501[i];
    if e24501[i]>0 and e38202[i]=4 and e38211[i]>=0 then ot[i]=e38211[i]/(2*e24501[i]);
    if e24501[i]>0 and e38202[i]=5 and e38211[i]>=0 then ot[i]=e38211[i]/(4.3*e24501[i]);
    if e24501[i]>0 and e38202[i]=6 and e38211[i]>=0 then ot[i]=-3; /* since no weeks per year available. */
    if e24501[i]>0 and e38202[i]=8 and e38211[i]>=0 then ot[i]=e38211[i]/(2.15*e24501[i]);
    if e38202[i] in (9,14) then ot[i]=0;
    if e38202[i] in (12,13) and e38211[i]>=0 then do;
/* condition based on the time unit to finish per ot. */
    if e38211aaa[i]=1 and e38211aab[i]>0 then ot[i]=(e38211[i]/e38211aab[i])*60;

```

Appendix 2: Employment Variable Creation

```

if e38211aaa[i]=2 and e38211aab[i]>0 then ot[i]=e38211[i]/e38211aab[i];
if e38211aaa[i]=3 and e38211aab[i]>0 and e38211b[i]>=0 and e24501[i]>0 then
    ot[i]=(e38211[i]/e38211aab[i])*e38211b[i]/e24501[i];
if e38211aaa[i]=4 and e38211aab[i]>0 and e24501[i]>0 then ot[i]=(e38211[i]/e38211aab[i])/e24501[i];
if e38211aaa[i]=5 and e38211aab[i]>0 and e24501[i]>0 then ot[i]=(e38211[i]/e38211aab[i])/(e24501[i]*4.3);
/* missing value */
if -4<e38211aaa[i]<0 or -4<e38211aab[i]<=0 then ot[i]=-3;
if e38211aaa[i] in (3,4,5) and -4<e24501[i]<=0 then ot[i]=-3;
if e38211aaa[i]=3 and e38211b[i]<0 then ot[i]=-3;
end;

if -4<e24501[i]<0 then ot[i]=e24501[i];
if e38211[i]>=0 and e24501[i]=0 then ot[i]=-3;
if -4<e38211[i]<0 then ot[i]=e38211[i];
if e38202[i]=2 and e38211b[i] le 0 then ot[i]=-3;
if e38202[i]=2 and -4<e38211b[i]<0 then ot[i]=e38211b[i];

end;

if e19200[i] ne 1 then do; /*report payment in other units at the begining*/
  if e34403[i]>0 and e38202[i]=1 and e38211[i]>=0 then ot[i]=e38211[i];
  if e34403[i]>0 and e38202[i]=2 and e38211b[i]>0 and e38211[i]>=0 then ot[i]=e38211[i]*e38211b[i]/e34403[i];
  if e34403[i]>0 and e38202[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38211[i]>=0
    then ot[i]=e38211[i]/e34403[i];
  if e34403[i]>0 and e38202[i]=4 and e38211[i]>=0 then ot[i]=e38211[i]/(2*e34403[i]);
  if e34403[i]>0 and e38202[i]=5 and e38211[i]>=0 then ot[i]=e38211[i]/(4.3*e34403[i]);
  if e34403[i]>0 and e38202[i]=6 and e38211[i]>=0 then ot[i]=-3; /* since no weeks per year available. */
  if e34403[i]>0 and e38202[i]=8 and e38211[i]>=0 then ot[i]=e38211[i]/(2.15*e34403[i]);
  if e38202[i] in (9,14) then ot[i]=0;
  if e38202[i] in (12,13) and e38211[i]>=0 then do;
    /* condition based on the time unit to finish per ot. */
    if e38211aaa[i]=1 and e38211aab[i]>0 then ot[i]=(e38211[i]/e38211aab[i])*60;
    if e38211aaa[i]=2 and e38211aab[i]>0 then ot[i]=e38211[i]/e38211aab[i];
    if e38211aaa[i]=3 and e38211aab[i]>0 and e38211b[i]>=0 and e34403[i]>0 then
      ot[i]=(e38211[i]/e38211aab[i])*e38211b[i]/e34403[i];
    if e38211aaa[i]=4 and e38211aab[i]>0 and e34403[i]>0 then ot[i]=(e38211[i]/e38211aab[i])/e34403[i];
    if e38211aaa[i]=5 and e38211aab[i]>0 and e34403[i]>0 then ot[i]=(e38211[i]/e38211aab[i])/(e34403[i]*4.3);
  /* missing value */
  if -4<e38211aaa[i]<0 or -4<e38211aab[i]<=0 then ot[i]=-3;
  if e38211aaa[i] in (3,4,5) and -4<e34403[i]<=0 then ot[i]=-3;
  if e38211aaa[i]=3 and e38211b[i]<0 then ot[i]=-3;
end;

if -4<e34403[i]<0 then ot[i]=e34403[i];
if e38211[i]>=0 and e34403[i]=0 then ot[i]=-3;
if -4<e38211[i]<0 then ot[i]=e38211[i];
if e38202[i]=2 and e38211b[i] le 0 then ot[i]=-3;
if e38202[i]=2 and -4<e38211b[i]<0 then ot[i]=e38211b[i];

end;

else do; /* different no. of hours from the beginning*/
  if e38105[i]>0 and e38202[i]=1 and e38211[i]>=0 then ot[i]=e38211[i];
  if e38105[i]>0 and e38202[i]=2 and e38211b[i]>0 and e38211[i]>=0 then ot[i]=e38211[i]*e38211b[i]/e38105[i];

```

```

if e38105[i]>0 and e38202[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38211[i]>=0
    then ot[i]=e38211[i]/e38105[i];
if e38105[i]>0 and e38202[i]=4 and e38211[i]>=0 then ot[i]=e38211[i]/(2*e38105[i]);
if e38105[i]>0 and e38202[i]=5 and e38211[i]>=0 then ot[i]=e38211[i]/(4.3*e38105[i]);
if e38105[i]>0 and e38202[i]=6 and e38211[i]>=0 then ot[i]=-3; /* snice no weeks per year available. */
if e38105[i]>0 and e38202[i]=8 and e38211[i]>=0 then ot[i]=e38211[i]/(2.15*e38105[i]);
if e38202[i] in (9,14) then ot[i]=0;
if e38202[i] in (12,13) and e38211[i]>=0 then do;
/* condition based on the time unit to finish per ot. */
    if e38211aaa[i]=1 and e38211aab[i]>0 then ot[i]=(e38211[i]/e38211aab[i])*60;
    if e38211aaa[i]=2 and e38211aab[i]>0 then ot[i]=e38211[i]/e38211aab[i];
    if e38211aaa[i]=3 and e38211aab[i]>0 and e38211b[i]>=0 and e38211[i]>0 then
        ot[i]=(e38211[i]/e38211aab[i])*e38211b[i]/e38105[i];
    if e38211aaa[i]=4 and e38211aab[i]>0 and e38211[i]>0 then ot[i]=(e38211[i]/e38211aab[i])/e38105[i];
    if e38211aaa[i]=5 and e38211aab[i]>0 and e38211[i]>0 then ot[i]=(e38211[i]/e38211aab[i])/(e38105[i]*4.3);
/* missing value */
    if -4<e38211aaa[i]<0 or -4<e38211aab[i]<=0 then ot[i]=-3;
    if e38211aaa[i] in (3,4,5) and -4<e38211[i]<=0 then ot[i]=-3;
    if e38211aaa[i]=3 and e38211b[i]<0 then ot[i]=-3;
end;

if -4<e38105[i]<0 then ot[i]=e38105[i];
if e38211[i]>=0 and e38105[i]=0 then ot[i]=-3;
if -4<e38211[i]<0 then ot[i]=e38211[i];
if e38202[i]=2 and e38211b[i] le 0 then ot[i]=-3;
if e38202[i]=2 and -4<e38211b[i]<0 then ot[i]=e38211b[i];

end;

end;
end;
end;

***** non-overtime payment *****
do i=1 to 9;

if (e37901b[i]=1 or e59900[i]=1) then do;

/** case i. without overtime at the beginning, same no. of hours. ***/
if e3800b[i]=1 and e212001[i] ne 1 then do;

/* report hourly wage at the beginning*/
/* for tips*/

if e23901[i]>0 and e384071[i]=1 and e384161[i]>=0 then othpay1[i]=e384161[i];
if e23901[i]>0 and e384071[i]=2 and e38416k1[i]>0 and e384161[i]>=0 then
    othpay1[i]=(e384161[i]*e38416k1[i])/e23901[i];
if e23901[i]>0 and e384071[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384161[i]>=0
    then othpay1[i]=e384161[i]/e23901[i];
if e23901[i]>0 and e384071[i]=4 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2*e23901[i]);
if e23901[i]>0 and e384071[i]=5 and e384161[i]>=0 then othpay1[i]=e384161[i]/(4.3*e23901[i]);
if e23901[i]>0 and e384071[i]=6 and e384161[i]>=0 and e35600[i]>0 then
    othpay1[i]=e384161[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e384071[i]=8 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2.15*e23901[i]);
if e384071[i] in (9,14) then othpay1[i]=0;
if e384071[i] in (12,13) and e384161[i]>=0 then do;
    if e38416g1[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */

```

Appendix 2: Employment Variable Creation

```

if e225301[i] in (1,2,4,5) and e225001[i]>0 and othpay1[i]>=0 then
    othpay1[i]=othpay1[i]/e225001[i]*e384161[i];
if e225301[i]=3 and e225001[i]>0 and e38416k1[i]>0 and e225501[i]>0 and othpay1[i]>=0
    then othpay1[i]=othpay1[i]/(e225001[i]*e225501[i])*e384161[i]*e38416k1[i];
if e225301[i]=3 and e38416k1[i]<=0 then othpay1[i]=-3;
end;
else do; /*in all the other cases */
if e38416i1[i]=1 and e38416ib1[i]>0 then othpay1[i]=(e384161[i]/e38416ib1[i])*60;
if e38416i1[i]=2 and e38416ib1[i]>0 then othpay1[i]=e384161[i]/e38416ib1[i];
if e38416i1[i]=3 and e38416ib1[i]>0 and e38416k1[i]>=0 and e23901[i]>0 then
    othpay1[i]=(e384161[i]/e38416ib1[i])*e38416k1[i]/e23901[i];
if e38416i1[i]=4 and e38416ib1[i]>0 and e23901[i]>0 then othpay1[i]=(e384161[i]/e38416ib1[i])/e23901[i];
if e38416i1[i]=5 and e38416ib1[i]>0 and e23901[i]>0 then othpay1[i]=(e384161[i]/e38416ib1[i])/(e23901[i]*4.3);
/* missing value */
if -4<e38416i1[i]<0 or -4<e38416ib1[i]<=0 then othpay1[i]=-3;
if e38416i1[i] in (3,4,5) and -4<e23901[i]<=0 then othpay1[i]=-3;
if e38416i1[i]=3 and e38416k1[i]<0 then othpay1[i]=-3;
end;
end;

if e384071[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e384161[i]<0 then
    othpay1[i]=e225001[i];
if e384071[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay1[i]=-3;
if e384071[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay1[i]=e23901[i];
if e384071[i]=2 and e38416k1[i] le 0 then othpay1[i]=-3;
if e384071[i]=2 and -4<e38416k1[i]<0 then othpay1[i]=e225501[i];
if e384071[i]=6 and e35600[i] le 0 then othpay1[i]=-3;
if e384071[i]=6 and -4<e35600[i]<0 then othpay1[i]=e35600[i];

/*for commissions*/
if e23901[i]>0 and e384072[i]=1 and e384162[i]>=0 then othpay2[i]=e384162[i];
if e23901[i]>0 and e384072[i]=2 and e38416k2[i]>0 and e384162[i]>=0 then
    othpay2[i]=(e384162[i]*e38416k2[i])/e23901[i];
if e23901[i]>0 and e384072[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999,-1,-2) and e384162[i]>=0
    then othpay2[i]=e384162[i]/e23901[i];
if e23901[i]>0 and e384072[i]=4 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2*e23901[i]);
if e23901[i]>0 and e384072[i]=5 and e384162[i]>=0 then othpay2[i]=e384162[i]/(4.3*e23901[i]);
if e23901[i]>0 and e384072[i]=6 and e384162[i]>=0 and e35600[i]>0 then
    othpay2[i]=e384162[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e384072[i]=8 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2.15*e23901[i]);
if e384072[i] in (9,14) then othpay2[i]=0;
if e384072[i] in (12,13) and e384162[i]>=0 then do;
    if e38416g2[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225302[i] in (1,2,4,5) and e225002[i]>0 and othpay2[i]>=0 then
            othpay2[i]=othpay2[i]/e225002[i]*e384162[i];
        if e225302[i]=3 and e225002[i]>0 and e38416k2[i]>0 and e225502[i]>0 and othpay2[i]>=0
            then othpay2[i]=othpay2[i]/(e225002[i]*e225502[i])*e384162[i]*e38416k2[i];
        if e225302[i]=3 and e38416k2[i]<=0 then othpay2[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38416i2[i]=1 and e38416ib2[i]>0 then othpay2[i]=(e384162[i]/e38416ib2[i])*60;
        if e38416i2[i]=2 and e38416ib2[i]>0 then othpay2[i]=e384162[i]/e38416ib2[i];
        if e38416i2[i]=3 and e38416ib2[i]>0 and e38416k2[i]>=0 and e23901[i]>0 then
            othpay2[i]=(e384162[i]/e38416ib2[i])*e38416k2[i]/e23901[i];
        if e38416i2[i]=4 and e38416ib2[i]>0 and e23901[i]>0 then othpay2[i]=(e384162[i]/e38416ib2[i])/e23901[i];

```

Appendix 2: Employment Variable Creation

```

if e38416i2[i]=5 and e38416ib2[i]>0 and e23901[i]>0 then othpay2[i]=(e384162[i]/e38416ib2[i])/(e23901[i]*4.3);
/* missing value */
if -4<e38416i2[i]<0 or -4<e38416ib2[i]<=0 then othpay2[i]=-3;
if e38416i2[i] in (3,4,5) and -4<e23901[i]<=0 then othpay2[i]=-3;
if e38416i2[i]=3 and e38416k2[i]<0 then othpay2[i]=-3;
end;
end;

if e384072[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e384162[i]<0 then
    othpay2[i]=e225002[i];
if e384072[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay2[i]=-3;
if e384072[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay2[i]=e23901[i];
if e384072[i]=2 and e38416k2[i]<0 then othpay2[i]=-3;
if e384072[i]=2 and -4<e38416k2[i]<0 then othpay2[i]=e225502[i];
if e384072[i]=6 and e35600[i] le 0 then othpay2[i]=-3;
if e384072[i]=6 and -4<e35600[i]<0 then othpay2[i]=e35600[i];

/*for bonuses*/

if e23901[i]>0 and e384073[i]=1 and e384163[i]>=0 then othpay3[i]=e384163[i];
if e23901[i]>0 and e384073[i]=2 and e38416k3[i]>0 and e384163[i]>=0 then
    othpay3[i]=(e384163[i]*e38416k3[i])/e23901[i];
if e23901[i]>0 and e384073[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384163[i]>=0
    then othpay3[i]=e384163[i]/e23901[i];
if e23901[i]>0 and e384073[i]=4 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2*e23901[i]);
if e23901[i]>0 and e384073[i]=5 and e384163[i]>=0 then othpay3[i]=e384163[i]/(4.3*e23901[i]);
if e23901[i]>0 and e384073[i]=6 and e384163[i]>=0 and e35600[i]>0 then
    othpay3[i]=e384163[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e384073[i]=8 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2.15*e23901[i]);
if e384073[i] in (9,14) then othpay3[i]=0;
if e384073[i] in (12,13) and e384163[i]>=0 then do;
    if e38416g3[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225303[i] in (1,2,4,5) and e225003[i]>0 and othpay3[i]>=0 then
            othpay3[i]=othpay3[i]/e225003[i]*e384163[i];
        if e225303[i]=3 and e225003[i]>0 and e38416k3[i]>0 and e225503[i]>0 and othpay3[i]>=0
            then othpay3[i]=othpay3[i]/(e225003[i]*e225503[i])*e384163[i]*e38416k3[i];
        if e225303[i]=3 and e38416k3[i]<=0 then othpay3[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38416i3[i]=1 and e38416ib3[i]>0 then othpay3[i]=(e384163[i]/e38416ib3[i])*60;
        if e38416i3[i]=2 and e38416ib3[i]>0 then othpay3[i]=e384163[i]/e38416ib3[i];
        if e38416i3[i]=3 and e38416ib3[i]>0 and e38416k3[i]>=0 and e23901[i]>0 then
            othpay3[i]=(e384163[i]/e38416ib3[i])*e38416k3[i]/e23901[i];
        if e38416i3[i]=4 and e38416ib3[i]>0 and e23901[i]>0 then othpay3[i]=(e384163[i]/e38416ib3[i])/e23901[i];
        if e38416i3[i]=5 and e38416ib3[i]>0 and e23901[i]>0 then othpay3[i]=(e384163[i]/e38416ib3[i])/(e23901[i]*4.3);
    /* missing value */
        if -4<e38416i3[i]<0 or -4<e38416ib3[i]<=0 then othpay3[i]=-3;
        if e38416i3[i] in (3,4,5) and -4<e23901[i]<=0 then othpay3[i]=-3;
        if e38416i3[i]=3 and e38416k3[i]<0 then othpay3[i]=-3;
    end;
end;

if e384073[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e384163[i]<0 then
    othpay3[i]=e225003[i];
if e384073[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay3[i]=-3;

```

Appendix 2: Employment Variable Creation

```

if e384073[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay3[i]=e23901[i];
if e384073[i]=2 and e38416k3[i] le 0 then othpay3[i]=-3;
if e384073[i]=2 and -4<e38416k3[i]<0 then othpay3[i]=e225503[i];
if e384073[i]=6 and e35600[i] le 0 then othpay3[i]=-3;
if e384073[i]=6 and -4<e35600[i]<0 then othpay3[i]=e35600[i];

/*for incentive pay*/

if e23901[i]>0 and e384074[i]=1 and e384164[i]>=0 then othpay4[i]=e384164[i];
if e23901[i]>0 and e384074[i]=2 and e38416k4[i]>0 and e384164[i]>=0 then
    othpay4[i]=(e384164[i]*e38416k4[i])/e23901[i];
if e23901[i]>0 and e384074[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384164[i]>=0
    then othpay4[i]=e384164[i]/e23901[i];
if e23901[i]>0 and e384074[i]=4 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2*e23901[i]);
if e23901[i]>0 and e384074[i]=5 and e384164[i]>=0 then othpay4[i]=e384164[i]/(4.3*e23901[i]);
if e23901[i]>0 and e384074[i]=6 and e384164[i]>=0 and e35600[i]>0 then
    othpay4[i]=e384164[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e384074[i]=8 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2.15*e23901[i]);
if e384074[i] in (9,14) then othpay4[i]=0;
if e384074[i] in (12,13) and e384164[i]>=0 then do;
    if e38416g4[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225304[i] in (1,2,4,5) and e225004[i]>0 and othpay4[i]>=0 then
            othpay4[i]=othpay4[i]/e225004[i]*e384164[i];
        if e225304[i]=3 and e225004[i]>0 and e38416k4[i]>0 and e225504[i]>0 and othpay4[i]>=0
            then othpay4[i]=othpay4[i]/(e225004[i]*e225504[i])*e384164[i]*e38416k4[i];
        if e225304[i]=3 and e38416k4[i]<=0 then othpay4[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38416i4[i]=1 and e38416ib4[i]>0 then othpay4[i]=(e384164[i]/e38416ib4[i])*60;
        if e38416i4[i]=2 and e38416ib4[i]>0 then othpay4[i]=e384164[i]/e38416ib4[i];
        if e38416i4[i]=3 and e38416ib4[i]>0 and e38416k4[i]>=0 and e23901[i]>0 then
            othpay4[i]=(e384164[i]/e38416ib4[i])*e38416k4[i]/e23901[i];
        if e38416i4[i]=4 and e38416ib4[i]>0 and e23901[i]>0 then othpay4[i]=(e384164[i]/e38416ib4[i])/e23901[i];
        if e38416i4[i]=5 and e38416ib4[i]>0 and e23901[i]>0 then othpay4[i]=(e384164[i]/e38416ib4[i])/(e23901[i]*4.3);
    /* missing value */
        if -4<e38416i4[i]<0 or -4<e38416ib4[i]<=0 then othpay4[i]=-3;
        if e38416i4[i] in (3,4,5) and -4<e23901[i]<=0 then othpay4[i]=-3;
        if e38416i4[i]=3 and e38416k4[i]<0 then othpay4[i]=-3;
    end;
end;
if e384074[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e384164[i]<0 then
    othpay4[i]=e225004[i];
if e384074[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay4[i]=-3;
if e384074[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay4[i]=e23901[i];
if e384074[i]=2 and e38416k4[i] le 0 then othpay4[i]=-3;
if e384074[i]=2 and -4<e38416k4[i]<0 then othpay4[i]=e225504[i];
if e384074[i]=6 and e35600[i] le 0 then othpay4[i]=-3;
if e384074[i]=6 and -4<e35600[i]<0 then othpay4[i]=e35600[i];

/*for others*/

if e23901[i]>0 and e384075[i]=1 and e384165[i]>=0 then othpay5[i]=e384165[i];
if e23901[i]>0 and e384075[i]=2 and e38416k5[i]>0 and e384165[i]>=0 then
    othpay5[i]=(e384165[i]*e38416k5[i])/e23901[i];

```

```

if e23901[i]>0 and e384075[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384165[i]>=0
    then othpay5[i]=e384165[i]/e23901[i];
if e23901[i]>0 and e384075[i]=4 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2*e23901[i]);
if e23901[i]>0 and e384075[i]=5 and e384165[i]>=0 then othpay5[i]=e384165[i]/(4.3*e23901[i]);
if e23901[i]>0 and e384075[i]=6 and e384165[i]>=0 and e35600[i]>0 then
    othpay5[i]=e384165[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e384075[i]=8 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2.15*e23901[i]);
if e384075[i] in (9,14) then othpay5[i]=0;
if e384075[i] in (12,13) and e384165[i]>=0 then do;
    if e38416g5[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225305[i] in (1,2,4,5) and e225005[i]>0 and othpay5[i]>=0 then
            othpay5[i]=othpay5[i]/e225005[i]*e384165[i];
        if e225305[i]=3 and e225005[i]>0 and e38416k5[i]>0 and e225505[i]>0 and othpay5[i]>=0
            then othpay5[i]=othpay5[i]/(e225005[i]*e225505[i])*e384165[i]*e38416k5[i];
        if e225305[i]=3 and e38416k5[i]<=0 then othpay5[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38416i5[i]=1 and e38416ib5[i]>0 then othpay5[i]=(e384165[i]/e38416ib5[i])*60;
        if e38416i5[i]=2 and e38416ib5[i]>0 then othpay5[i]=e384165[i]/e38416ib5[i];
        if e38416i5[i]=3 and e38416ib5[i]>0 and e38416k5[i]>0 and e23901[i]>0 then
            othpay5[i]=(e384165[i]/e38416ib5[i])*e38416k5[i]/e23901[i];
        if e38416i5[i]=4 and e38416ib5[i]>0 and e23901[i]>0 then othpay5[i]=(e384165[i]/e38416ib5[i])/e23901[i];
        if e38416i5[i]=5 and e38416ib5[i]>0 and e23901[i]>0 then othpay5[i]=(e384165[i]/e38416ib5[i])/(e23901[i]*4.3);
    /* missing value */
        if -4<e38416i5[i]<0 or -4<e38416ib5[i]<=0 then othpay5[i]=-3;
        if e38416i5[i] in (3,4,5) and -4<e23901[i]<=0 then othpay5[i]=-3;
        if e38416i5[i]=3 and e38416k5[i]<0 then othpay5[i]=-3;
    end;
end;
if e384075[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e384165[i]<0 then
    othpay5[i]=e225005[i];
if e384075[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay5[i]=-3;
if e384075[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay5[i]=e23901[i];
if e384075[i]=2 and e38416k5[i]<0 then othpay5[i]=-3;
if e384075[i]=2 and -4<e38416k5[i]<0 then othpay5[i]=e225505[i];
if e384075[i]=6 and e35600[i]<0 then othpay5[i]=-3;
if e384075[i]=6 and -4<e35600[i]<0 then othpay5[i]=e35600[i];

/* report nonhourly wage at the beginning*/
/* for tips*/
if e34402[i]>0 and e384071[i]=1 and e384161[i]>=0 then othpay1[i]=e384161[i];
if e34402[i]>0 and e384071[i]=2 and e38416k1[i]>0 and e384161[i]>=0 then
    othpay1[i]=(e384161[i]*e38416k1[i])/e34402[i];
if e34402[i]>0 and e384071[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384161[i]>=0
    then othpay1[i]=e384161[i]/e34402[i];
if e34402[i]>0 and e384071[i]=4 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2*e34402[i]);
if e34402[i]>0 and e384071[i]=5 and e384161[i]>=0 then othpay1[i]=e384161[i]/(4.3*e34402[i]);
if e34402[i]>0 and e384071[i]=6 and e384161[i]>=0 and e35600[i]>0 then
    othpay1[i]=e384161[i]/(e35600[i]*e34402[i]);
if e34402[i]>0 and e384071[i]=8 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2.15*e34402[i]);
if e384071[i] in (12,13) and e384161[i]>=0 then do;
    if e38416i1[i]=3 and e38416ib1[i]>0 and e38416k1[i]>=0 and e34402[i]>0 then
        othpay1[i]=(e384161[i]/e38416ib1[i])*e38416k1[i]/e34402[i];

```

Appendix 2: Employment Variable Creation

```
if e38416i1[i]=4 and e38416ib1[i]>0 and e34402[i]>0 then othpay1[i]=(e384161[i]/e38416ib1[i])/e34402[i];
if e38416i1[i]=5 and e38416ib1[i]>0 and e34402[i]>0 then othpay1[i]=(e384161[i]/e38416ib1[i])/(e34402[i]*4.3);
/* missing value */
if e38416i1[i] in (3,4,5) and -4<e34402[i]<=0 then othpay1[i]=-3;
end;

if e384071[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay1[i]=-3;
if e384071[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
    othpay1[i]=e34402[i];

/*for commissions*/
if e34402[i]>0 and e384072[i]=1 and e384162[i]>=0 then othpay2[i]=e384162[i];
if e34402[i]>0 and e384072[i]=2 and e38416k2[i]>0 and e384162[i]>=0 then
    othpay2[i]=(e384162[i]*e38416k2[i])/e34402[i];
if e34402[i]>0 and e384072[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384162[i]>=0
    then othpay2[i]=e384162[i]/e34402[i];
if e34402[i]>0 and e384072[i]=4 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2*e34402[i]);
if e34402[i]>0 and e384072[i]=5 and e384162[i]>=0 then othpay2[i]=e384162[i]/(4.3*e34402[i]);
if e34402[i]>0 and e384072[i]=6 and e384162[i]>=0 and e35600[i]>0 then
    othpay2[i]=e384162[i]/(e35600[i]*e34402[i]);
if e34402[i]>0 and e384072[i]=8 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2.15*e34402[i]);
if e384072[i] in (12,13) and e384162[i]>=0 then do;
    if e38416i2[i]=3 and e38416ib2[i]>0 and e38416k2[i]>=0 and e34402[i]>0 then
        othpay2[i]=(e384162[i]/e38416ib2[i])*e38416k2[i]/e34402[i];
    if e38416i2[i]=4 and e38416ib2[i]>0 and e34402[i]>0 then othpay2[i]=(e384162[i]/e38416ib2[i])/e34402[i];
    if e38416i2[i]=5 and e38416ib2[i]>0 and e34402[i]>0 then othpay2[i]=(e384162[i]/e38416ib2[i])/(e34402[i]*4.3);
/* missing value */
    if e38416i2[i] in (3,4,5) and -4<e34402[i]<=0 then othpay2[i]=-3;
end;

if e384072[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay2[i]=-3;
if e384072[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
    othpay2[i]=e34402[i];

/*for bonuses*/
if e34402[i]>0 and e384073[i]=1 and e384163[i]>=0 then othpay3[i]=e384163[i];
if e34402[i]>0 and e384073[i]=2 and e38416k3[i]>0 and e384163[i]>=0 then
    othpay3[i]=(e384163[i]*e38416k3[i])/e34402[i];
if e34402[i]>0 and e384073[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384163[i]>=0
    then othpay3[i]=e384163[i]/e34402[i];
if e34402[i]>0 and e384073[i]=4 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2*e34402[i]);
if e34402[i]>0 and e384073[i]=5 and e384163[i]>=0 then othpay3[i]=e384163[i]/(4.3*e34402[i]);
if e34402[i]>0 and e384073[i]=6 and e384163[i]>=0 and e35600[i]>0 then
    othpay3[i]=e384163[i]/(e35600[i]*e34402[i]);
if e34402[i]>0 and e384073[i]=8 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2.15*e34402[i]);
if e384073[i] in (12,13) and e384163[i]>=0 then do;
    if e38416i3[i]=3 and e38416ib3[i]>0 and e38416k3[i]>=0 and e34402[i]>0 then
        othpay3[i]=(e384163[i]/e38416ib3[i])*e38416k3[i]/e34402[i];
    if e38416i3[i]=4 and e38416ib3[i]>0 and e34402[i]>0 then othpay3[i]=(e384163[i]/e38416ib3[i])/e34402[i];
    if e38416i3[i]=5 and e38416ib3[i]>0 and e34402[i]>0 then othpay3[i]=(e384163[i]/e38416ib3[i])/(e34402[i]*4.3);
/* missing value */
    if e38416i3[i] in (3,4,5) and -4<e34402[i]<=0 then othpay3[i]=-3;
end;

if e384073[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay3[i]=-3;
```

Appendix 2: Employment Variable Creation

```

if e384073[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
    othpay3[i]=e34402[i];

/*for incentive pay*/

if e34402[i]>0 and e384074[i]=1 and e384164[i]>=0 then othpay4[i]=e384164[i];
if e34402[i]>0 and e384074[i]=2 and e38416k4[i]>0 and e384164[i]>=0 then
    othpay4[i]=(e384164[i]*e38416k4[i])/e34402[i];
if e34402[i]>0 and e384074[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384164[i]>=0
    then othpay4[i]=e384164[i]/e34402[i];
if e34402[i]>0 and e384074[i]=4 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2*e34402[i]);
if e34402[i]>0 and e384074[i]=5 and e384164[i]>=0 then othpay4[i]=e384164[i]/(4.3*e34402[i]);
if e34402[i]>0 and e384074[i]=6 and e384164[i]>=0 and e35600[i]>0 then
    othpay4[i]=e384164[i]/(e35600[i]*e34402[i]);
if e34402[i]>0 and e384074[i]=8 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2.15*e34402[i]);
if e384074[i] in (12,13) and e384164[i]>=0 then do;
    if e38416i4[i]=3 and e38416ib4[i]>0 and e38416k4[i]>=0 and e34402[i]>0 then
        othpay4[i]=(e384164[i]/e38416ib4[i])*e38416k4[i]/e34402[i];
    if e38416i4[i]=4 and e38416ib4[i]>0 and e34402[i]>0 then othpay4[i]=(e384164[i]/e38416ib4[i])/e34402[i];
    if e38416i4[i]=5 and e38416ib4[i]>0 and e34402[i]>0 then othpay4[i]=(e384164[i]/e38416ib4[i])/(e34402[i]*4.3);
    /* missing value */
    if e38416i4[i] in (3,4,5) and -4<e34402[i]<=0 then othpay4[i]=-3;
end;

if e384074[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay4[i]=-3;
if e384074[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
    othpay4[i]=e34402[i];

/*for others*/

if e34402[i]>0 and e384075[i]=1 and e384165[i]>=0 then othpay5[i]=e384165[i];
if e34402[i]>0 and e384075[i]=2 and e38416k5[i]>0 and e384165[i]>=0 then
    othpay5[i]=(e384165[i]*e38416k5[i])/e34402[i];
if e34402[i]>0 and e384075[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384165[i]>=0
    then othpay5[i]=e384165[i]/e34402[i];
if e34402[i]>0 and e384075[i]=4 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2*e34402[i]);
if e34402[i]>0 and e384075[i]=5 and e384165[i]>=0 then othpay5[i]=e384165[i]/(4.3*e34402[i]);
if e34402[i]>0 and e384075[i]=6 and e384165[i]>=0 and e35600[i]>0 then
    othpay5[i]=e384165[i]/(e35600[i]*e34402[i]);
if e34402[i]>0 and e384075[i]=8 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2.15*e34402[i]);
if e384075[i] in (12,13) and e384165[i]>=0 then do;
    if e38416i5[i]=3 and e38416ib5[i]>0 and e38416k5[i]>=0 and e34402[i]>0 then
        othpay5[i]=(e384165[i]/e38416ib5[i])*e38416k5[i]/e34402[i];
    if e38416i5[i]=4 and e38416ib5[i]>0 and e34402[i]>0 then othpay5[i]=(e384165[i]/e38416ib5[i])/e34402[i];
    if e38416i5[i]=5 and e38416ib5[i]>0 and e34402[i]>0 then othpay5[i]=(e384165[i]/e38416ib5[i])/(e34402[i]*4.3);
    /* missing value */
    if e38416i5[i] in (3,4,5) and -4<e34402[i]<=0 then othpay5[i]=-3;
end;

if e384075[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay5[i]=-3;
if e384075[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
    othpay5[i]=e34402[i];

end;

/** case ii. without overtime at the beginning, diff no. of hours.***/

```

Appendix 2: Employment Variable Creation

```

if e3800b[i]=0 and e212001[i] ne 1 then do;
/* for tips*/
  if e3800ff[i]>0 and e384071[i]=1 and e384161[i]>=0 then othpay1[i]=e384161[i];
  if e3800ff[i]>0 and e384071[i]=2 and e38416k1[i]>0 and e384161[i]>=0 then
    othpay1[i]=(e384161[i]*e38416k1[i])/e3800ff[i];
  if e3800ff[i]>0 and e384071[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384161[i]>=0
    then othpay1[i]=e384161[i]/e3800ff[i];
  if e3800ff[i]>0 and e384071[i]=4 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2*e3800ff[i]);
  if e3800ff[i]>0 and e384071[i]=5 and e384161[i]>=0 then othpay1[i]=e384161[i]/(4.3*e3800ff[i]);
  if e3800ff[i]>0 and e384071[i]=6 and e384161[i]>=0 and e35600[i]>0 then
    othpay1[i]=e384161[i]/(e35600[i]*e3800ff[i]);
  if e3800ff[i]>0 and e384071[i]=8 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2.15*e3800ff[i]);
  if e384071[i] in (9,14) then othpay1[i]=0;
  if e384071[i] in (12,13) and e384161[i]>=0 then do;
    if e38416g1[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
      if e225301[i] in (1,2,4,5) and e225001[i]>0 and othpay1[i]>=0 then
        othpay1[i]=othpay1[i]/e225001[i]*e384161[i];
      if e225301[i]=3 and e225001[i]>0 and e38416k1[i]>0 and e225501[i]>0 and othpay1[i]>=0
        then othpay1[i]=othpay1[i]/(e225001[i]*e225501[i])*e384161[i]*e38416k1[i];
      if e225301[i]=3 and e38416k1[i]<=0 then othpay1[i]=-3;
    end;
    else do; /*in all the other cases */
      if e38416i1[i]=1 and e38416ib1[i]>0 then othpay1[i]=(e384161[i]/e38416ib1[i])*60;
      if e38416i1[i]=2 and e38416ib1[i]>0 then othpay1[i]=e384161[i]/e38416ib1[i];
      if e38416i1[i]=3 and e38416ib1[i]>0 and e38416k1[i]>=0 and e3800ff[i]>0 then
        othpay1[i]=(e384161[i]/e38416ib1[i])*e38416k1[i]/e3800ff[i];
      if e38416i1[i]=4 and e38416ib1[i]>0 and e3800ff[i]>0 then othpay1[i]=(e384161[i]/e38416ib1[i])/e3800ff[i];
      if e38416i1[i]=5 and e38416ib1[i]>0 and e3800ff[i]>0 then othpay1[i]=(e384161[i]/e38416ib1[i])/(e3800ff[i]*4.3);
    /* missing value */
      if -4<e38416i1[i]<0 or -4<e38416ib1[i]<=0 then othpay1[i]=-3;
      if e38416i1[i] in (3,4,5) and -4<e3800ff[i]<=0 then othpay1[i]=-3;
      if e38416i1[i]=3 and e38416k1[i]<0 then othpay1[i]=-3;
    end;
  end;
  if e384071[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e384161[i]<0 then
    othpay1[i]=e225001[i];
  if e384071[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e3800ff[i]=0 then othpay1[i]=-3;
  if e384071[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e3800ff[i]<0 then
    othpay1[i]=e3800ff[i];
  if e384071[i]=2 and e38416k1[i] le 0 then othpay1[i]=-3;
  if e384071[i]=2 and -4<e38416k1[i]<0 then othpay1[i]=e225501[i];
  if e384071[i]=6 and e35600[i] le 0 then othpay1[i]=-3;
  if e384071[i]=6 and -4<e35600[i]<0 then othpay1[i]=e35600[i];

/*for commissions*/
  if e3800ff[i]>0 and e384072[i]=1 and e384162[i]>=0 then othpay2[i]=e384162[i];
  if e3800ff[i]>0 and e384072[i]=2 and e38416k2[i]>0 and e384162[i]>=0 then
    othpay2[i]=(e384162[i]*e38416k2[i])/e3800ff[i];
  if e3800ff[i]>0 and e384072[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384162[i]>=0
    then othpay2[i]=e384162[i]/e3800ff[i];
  if e3800ff[i]>0 and e384072[i]=4 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2*e3800ff[i]);
  if e3800ff[i]>0 and e384072[i]=5 and e384162[i]>=0 then othpay2[i]=e384162[i]/(4.3*e3800ff[i]);

```

```

if e3800ff[i]>0 and e384072[i]=6 and e384162[i]>=0 and e35600[i]>0 then
    othpay2[i]=e384162[i]/(e35600[i]*e3800ff[i]);
if e3800ff[i]>0 and e384072[i]=8 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2.15*e3800ff[i]);
if e384072[i] in (9,14) then othpay2[i]=0;
if e384072[i] in (12,13) and e384162[i]>=0 then do;
    if e38416g2[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225302[i] in (1,2,4,5) and e225002[i]>0 and othpay2[i]>=0 then
            othpay2[i]=othpay2[i]/e225002[i]*e384162[i];
        if e225302[i]=3 and e225002[i]>0 and e38416k2[i]>0 and e225502[i]>0 and othpay2[i]>=0
            then othpay2[i]=othpay2[i]/(e225002[i]*e225502[i])*e384162[i]*e38416k2[i];
        if e225302[i]=3 and e38416k2[i]<=0 then othpay2[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38416i2[i]=1 and e38416ib2[i]>0 then othpay2[i]=(e384162[i]/e38416ib2[i])*60;
        if e38416i2[i]=2 and e38416ib2[i]>0 then othpay2[i]=e384162[i]/e38416ib2[i];
        if e38416i2[i]=3 and e38416ib2[i]>0 and e38416k2[i]>=0 and e3800ff[i]>0 then
            othpay2[i]=(e384162[i]/e38416ib2[i])*e38416k2[i]/e3800ff[i];
        if e38416i2[i]=4 and e38416ib2[i]>0 and e3800ff[i]>0 then othpay2[i]=(e384162[i]/e38416ib2[i])/e3800ff[i];
        if e38416i2[i]=5 and e38416ib2[i]>0 and e3800ff[i]>0 then othpay2[i]=(e384162[i]/e38416ib2[i])/(e3800ff[i]*4.3);
        /* missing value */
        if -4<e38416i2[i]<0 or -4<e38416ib2[i]<=0 then othpay2[i]=-3;
        if e38416i2[i] in (3,4,5) and -4<e3800ff[i]<=0 then othpay2[i]=-3;
        if e38416i2[i]=3 and e38416k2[i]<0 then othpay2[i]=-3;
    end;
end;
/*for bonuses*/
if e384072[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e384162[i]<0 then
    othpay2[i]=e225002[i];
if e384072[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e3800ff[i]=0 then othpay2[i]=-3;
if e384072[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e3800ff[i]<0 then
    othpay2[i]=e3800ff[i];
if e384072[i]=2 and e38416k2[i] le 0 then othpay2[i]=-3;
if e384072[i]=2 and -4<e38416k2[i]<0 then othpay2[i]=e225502[i];
if e384072[i]=6 and e35600[i] le 0 then othpay2[i]=-3;
if e384072[i]=6 and -4<e35600[i]<0 then othpay2[i]=e35600[i];

if e3800ff[i]>0 and e384073[i]=1 and e384163[i]>=0 then othpay3[i]=e384163[i];
if e3800ff[i]>0 and e384073[i]=2 and e38416k3[i]>0 and e384163[i]>=0 then
    othpay3[i]=(e384163[i]*e38416k3[i])/e3800ff[i];
if e3800ff[i]>0 and e384073[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384163[i]>=0
    then othpay3[i]=e384163[i]/e3800ff[i];
if e3800ff[i]>0 and e384073[i]=4 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2*e3800ff[i]);
if e3800ff[i]>0 and e384073[i]=5 and e384163[i]>=0 then othpay3[i]=e384163[i]/(4.3*e3800ff[i]);
if e3800ff[i]>0 and e384073[i]=6 and e384163[i]>=0 and e35600[i]>0 then
    othpay3[i]=e384163[i]/(e35600[i]*e3800ff[i]);
if e3800ff[i]>0 and e384073[i]=8 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2.15*e3800ff[i]);
if e384073[i] in (9,14) then othpay3[i]=0;
if e384073[i] in (12,13) and e384163[i]>=0 then do;
    if e38416g3[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225303[i] in (1,2,4,5) and e225003[i]>0 and othpay3[i]>=0 then
            othpay3[i]=othpay3[i]/e225003[i]*e384163[i];
        if e225303[i]=3 and e225003[i]>0 and e38416k3[i]>0 and e225503[i]>0 and othpay3[i]>=0
            then othpay3[i]=othpay3[i]/(e225003[i]*e225503[i])*e384163[i]*e38416k3[i];
        if e225303[i]=3 and e38416k3[i]<=0 then othpay3[i]=-3;
    end;

```

```

else do; /*in all the other cases */
if e38416i3[i]=1 and e38416ib3[i]>0 then othpay3[i]=(e384163[i]/e38416ib3[i])*60;
if e38416i3[i]=2 and e38416ib3[i]>0 then othpay3[i]=e384163[i]/e38416ib3[i];
if e38416i3[i]=3 and e38416ib3[i]>0 and e38416k3[i]>=0 and e3800ff[i]>0 then
    othpay3[i]=(e384163[i]/e38416ib3[i])*e38416k3[i]/e3800ff[i];
if e38416i3[i]=4 and e38416ib3[i]>0 and e3800ff[i]>0 then othpay3[i]=(e384163[i]/e38416ib3[i])/e3800ff[i];
if e38416i3[i]=5 and e38416ib3[i]>0 and e3800ff[i]>0 then othpay3[i]=(e384163[i]/e38416ib3[i])/(e3800ff[i]*4.3);
/* missing value */
if -4<e38416i3[i]<0 or -4<e38416ib3[i]<=0 then othpay3[i]=-3;
if e38416i3[i] in (3,4,5) and -4<e3800ff[i]<=0 then othpay3[i]=-3;
if e38416i3[i]=3 and e38416k3[i]<0 then othpay3[i]=-3;
end;
end;

if e384073[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e384163[i]<0 then
    othpay3[i]=e225003[i];
if e384073[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e3800ff[i]=0 then othpay3[i]=-3;
if e384073[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e3800ff[i]<0 then
    othpay3[i]=e3800ff[i];
if e384073[i]=2 and e38416k3[i] le 0 then othpay3[i]=-3;
if e384073[i]=2 and -4<e38416k3[i]<0 then othpay3[i]=e225503[i];
if e384073[i]=6 and e35600[i] le 0 then othpay3[i]=-3;
if e384073[i]=6 and -4<e35600[i]<0 then othpay3[i]=e35600[i];

/*for incentive pay*/
if e3800ff[i]>0 and e384074[i]=1 and e384164[i]>=0 then othpay4[i]=e384164[i];
if e3800ff[i]>0 and e384074[i]=2 and e38416k4[i]>0 and e384164[i]>=0 then
    othpay4[i]=(e384164[i]*e38416k4[i])/e3800ff[i];
if e3800ff[i]>0 and e384074[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384164[i]>=0
    then othpay4[i]=e384164[i]/e3800ff[i];
if e3800ff[i]>0 and e384074[i]=4 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2*e3800ff[i]);
if e3800ff[i]>0 and e384074[i]=5 and e384164[i]>=0 then othpay4[i]=e384164[i]/(4.3*e3800ff[i]);
if e3800ff[i]>0 and e384074[i]=6 and e384164[i]>=0 and e35600[i]>0 then
    othpay4[i]=e384164[i]/(e35600[i]*e3800ff[i]);
if e3800ff[i]>0 and e384074[i]=8 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2.15*e3800ff[i]);
if e384074[i] in (9,14) then othpay4[i]=0;
if e384074[i] in (12,13) and e384164[i]>=0 then do;
    if e38416g4[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225304[i] in (1,2,4,5) and e225004[i]>0 and othpay4[i]>=0 then
            othpay4[i]=othpay4[i]/e225004[i]*e384164[i];
        if e225304[i]=3 and e225004[i]>0 and e38416k4[i]>0 and e225504[i]>0 and othpay4[i]>=0
            then othpay4[i]=othpay4[i]/(e225004[i]*e225504[i])*e384164[i]*e38416k4[i];
        if e225304[i]=3 and e38416k4[i]<=0 then othpay4[i]=-3;
    end;
else do; /*in all the other cases */
    if e38416i4[i]=1 and e38416ib4[i]>0 then othpay4[i]=(e384164[i]/e38416ib4[i])*60;
    if e38416i4[i]=2 and e38416ib4[i]>0 then othpay4[i]=e384164[i]/e38416ib4[i];
    if e38416i4[i]=3 and e38416ib4[i]>0 and e38416k4[i]>=0 and e3800ff[i]>0 then
        othpay4[i]=(e384164[i]/e38416ib4[i])*e38416k4[i]/e3800ff[i];
    if e38416i4[i]=4 and e38416ib4[i]>0 and e3800ff[i]>0 then othpay4[i]=(e384164[i]/e38416ib4[i])/e3800ff[i];
    if e38416i4[i]=5 and e38416ib4[i]>0 and e3800ff[i]>0 then othpay4[i]=(e384164[i]/e38416ib4[i])/(e3800ff[i]*4.3);
/* missing value */
    if -4<e38416i4[i]<0 or -4<e38416ib4[i]<=0 then othpay4[i]=-3;
    if e38416i4[i] in (3,4,5) and -4<e3800ff[i]<=0 then othpay4[i]=-3;
    if e38416i4[i]=3 and e38416k4[i]<0 then othpay4[i]=-3;
end;

```

```

end;

if e384074[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e384164[i]<0 then
    othpay4[i]=e225004[i];
if e384074[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e3800ff[i]=0 then othpay4[i]=-3;
if e384074[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e3800ff[i]<0 then
    othpay4[i]=e3800ff[i];
if e384074[i]=2 and e38416k4[i] le 0 then othpay4[i]=-3;
if e384074[i]=2 and -4<e38416k4[i]<0 then othpay4[i]=e225504[i];
if e384074[i]=6 and e35600[i] le 0 then othpay4[i]=-3;
if e384074[i]=6 and -4<e35600[i]<0 then othpay4[i]=e35600[i];

/*for others*/

if e3800ff[i]>0 and e384075[i]=1 and e384165[i]>=0 then othpay5[i]=e384165[i];
if e3800ff[i]>0 and e384075[i]=2 and e38416k5[i]>0 and e384165[i]>=0 then
    othpay5[i]=(e384165[i]*e38416k5[i])/e3800ff[i];
if e3800ff[i]>0 and e384075[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384165[i]>=0
    then othpay5[i]=e384165[i]/e3800ff[i];
if e3800ff[i]>0 and e384075[i]=4 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2*e3800ff[i]);
if e3800ff[i]>0 and e384075[i]=5 and e384165[i]>=0 then othpay5[i]=e384165[i]/(4.3*e3800ff[i]);
if e3800ff[i]>0 and e384075[i]=6 and e384165[i]>=0 and e35600[i]>0 then
    othpay5[i]=e384165[i]/(e35600[i]*e3800ff[i]);
if e3800ff[i]>0 and e384075[i]=8 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2.15*e3800ff[i]);
if e384075[i] in (9,14) then othpay5[i]=0;
if e384075[i] in (12,13) and e384165[i]>=0 then do;
    if e38416g5[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225305[i] in (1,2,4,5) and e225005[i]>0 and othpay5[i]>=0 then
            othpay5[i]=othpay5[i]/e225005[i]*e384165[i];
        if e225305[i]=3 and e225005[i]>0 and e38416k5[i]>0 and e225505[i]>0 and othpay5[i]>=0
            then othpay5[i]=othpay5[i]/(e225005[i]*e225505[i])*e384165[i]*e38416k5[i];
        if e225305[i]=3 and e38416k5[i]<=0 then othpay5[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38416i5[i]=1 and e38416ib5[i]>0 then othpay5[i]=(e384165[i]/e38416ib5[i])*60;
        if e38416i5[i]=2 and e38416ib5[i]>0 then othpay5[i]=e384165[i]/e38416ib5[i];
        if e38416i5[i]=3 and e38416ib5[i]>0 and e38416k5[i]>=0 and e3800ff[i]>0 then
            othpay5[i]=(e384165[i]/e38416ib5[i])*e38416k5[i]/e3800ff[i];
        if e38416i5[i]=4 and e38416ib5[i]>0 and e3800ff[i]>0 then othpay5[i]=(e384165[i]/e38416ib5[i])/e3800ff[i];
        if e38416i5[i]=5 and e38416ib5[i]>0 and e3800ff[i]>0 then othpay5[i]=(e384165[i]/e38416ib5[i])/(e3800ff[i]*4.3);
    /* missing value */
        if -4<e38416i5[i]<0 or -4<e38416ib5[i]<=0 then othpay5[i]=-3;
        if e38416i5[i] in (3,4,5) and -4<e3800ff[i]<=0 then othpay5[i]=-3;
        if e38416i5[i]=3 and e38416k5[i]<0 then othpay5[i]=-3;
    end;
end;
if e384075[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e384165[i]<0 then
    othpay5[i]=e225005[i];
if e384075[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e3800ff[i]=0 then othpay5[i]=-3;
if e384075[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e3800ff[i]<0 then
    othpay5[i]=e3800ff[i];
if e384075[i]=2 and e38416k5[i] le 0 then othpay5[i]=-3;
if e384075[i]=2 and -4<e38416k5[i]<0 then othpay5[i]=e225505[i];
if e384075[i]=6 and e35600[i] le 0 then othpay5[i]=-3;
if e384075[i]=6 and -4<e35600[i]<0 then othpay5[i]=e35600[i];

```

Appendix 2: Employment Variable Creation

end;

/** case iii. only one compensation at the beginning,same no. of hours. ***/

if e38329[i]=1 and e20700[i]=1 then do;

/* for tips*/

if e212002[i]=1 then do;

/*with overtime at the beginning*/

if e38102[i] ne 1 and e38102[i] ne 3 then do;

/*report hourly wage at the beginning*/

if e23901[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay1[i]=e38329d[i];

if e23901[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
 othpay1[i]=(e38329d[i]*e38329k[i])/e23901[i];

if e23901[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
 then othpay1[i]=e38329d[i]/e23901[i];

if e23901[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2*e23901[i]);

if e23901[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(4.3*e23901[i]);

if e23901[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
 othpay1[i]=e38329d[i]/(e35600[i]*e23901[i]);

if e23901[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2.15*e23901[i]);

if e38329b[i] in (9,14) then othpay1[i]=0;

if e38329b[i] in (12,13) and e38329d[i]>=0 then do;

if e38329g[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */

if e225301[i] in (1,2,4,5) and e225001[i]>0 and othpay1[i]>=0 then
 othpay1[i]=othpay1[i]/e225001[i]*e38329d[i];

if e225301[i]=3 and e225001[i]>0 and e38329k[i]>0 and e225501[i]>0 and othpay1[i]>=0
 then othpay1[i]=othpay1[i]/(e225001[i]*e225501[i])*e38329d[i]*e38329k[i];

if e225301[i]=3 and e38329k[i]<=0 then othpay1[i]=-3;

end;

else do; /*in all the other cases */

if e38329i[i]=1 and e38329ib[i]>0 then othpay1[i]=(e38329d[i]/e38329ib[i])*60;

if e38329i[i]=2 and e38329ib[i]>0 then othpay1[i]=e38329d[i]/e38329ib[i];

if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>=0 and e23901[i]>0 then
 othpay1[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e23901[i];

if e38329i[i]=4 and e38329ib[i]>0 and e23901[i]>0 then othpay1[i]=(e38329d[i]/e38329ib[i])/e23901[i];

if e38329i[i]=5 and e38329ib[i]>0 and e23901[i]>0 then othpay1[i]=(e38329d[i]/e38329ib[i])/(e23901[i]*4.3);

/* missing value */

if -4<e38329i[i]<0 or -4<e38329ib[i]<=0 then othpay1[i]=-3;

if e38329i[i] in (3,4,5) and -4<e23901[i]<=0 then othpay1[i]=-3;

if e38329i[i]=3 and e38329k[i]<0 then othpay1[i]=-3;

end;

end;

if e38329b[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38329d[i]<0 then
 othpay1[i]=e225001[i];

if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay1[i]=-3;

if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
 othpay1[i]=e23901[i];

if e38329b[i]=2 and e38329k[i] le 0 then othpay1[i]=-3;

if e38329b[i]=2 and -4<e38329k[i]<0 then othpay1[i]=e225501[i];

if e38329b[i]=6 and e35600[i] le 0 then othpay1[i]=-3;

if e38329b[i]=6 and -4<e35600[i]<0 then othpay1[i]=e35600[i];

```

/*report nonhourly wage*/
if e34428[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay1[i]=e38329d[i];
if e34428[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay1[i]=(e38329d[i]*e38329k[i])/e34428[i];
if e34428[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay1[i]=e38329d[i]/e34428[i];
if e34428[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2*e34428[i]);
if e34428[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(4.3*e34428[i]);
if e34428[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay1[i]=e38329d[i]/(e35600[i]*e34428[i]);
if e34428[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2.15*e34428[i]);
if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
    if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>0 and e34428[i]>0 then
        othpay1[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e34428[i];
        if e38329i[i]=4 and e38329ib[i]>0 and e34428[i]>0 then othpay1[i]=(e38329d[i]/e38329ib[i])/e34428[i];
        if e38329i[i]=5 and e38329ib[i]>0 and e34428[i]>0 then othpay1[i]=(e38329d[i]/e38329ib[i])/(e34428[i]*4.3);
    /* missing value */
    if e38329i[i] in (3,4,5) and -4<e34428[i]<=0 then othpay1[i]=-3;
end;

if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34428[i]=0 then othpay1[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34428[i]<0 then
    othpay1[i]=e34428[i];

end;

/*without overtime at the beginning*/
if e3800b[i]=1 then do;

    /*report hourly wage at the beginning*/
    if e23901[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay1[i]=e38329d[i];
    if e23901[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
        othpay1[i]=(e38329d[i]*e38329k[i])/e23901[i];
    if e23901[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
        then othpay1[i]=e38329d[i]/e23901[i];
    if e23901[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2*e23901[i]);
    if e23901[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(4.3*e23901[i]);
    if e23901[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
        othpay1[i]=e38329d[i]/(e35600[i]*e23901[i]);
    if e23901[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2.15*e23901[i]);
    if e38329b[i] in (9,14) then othpay1[i]=0;
    if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
        if e38329g[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
            if e225301[i] in (1,2,4,5) and e225001[i]>0 and othpay1[i]>=0 then
                othpay1[i]=othpay1[i]/e225001[i]*e38329d[i];
            if e225301[i]=3 and e225001[i]>0 and e38329k[i]>0 and e225501[i]>0 and othpay1[i]>=0
                then othpay1[i]=othpay1[i]/(e225001[i]*e225501[i])*e38329d[i]*e38329k[i];
            if e225301[i]=3 and e38329k[i]<=0 then othpay1[i]=-3;
        end;
        else do; /*in all the other cases */
            if e38329i[i]=1 and e38329ib[i]>0 then othpay1[i]=(e38329d[i]/e38329ib[i])*60;
            if e38329i[i]=2 and e38329ib[i]>0 then othpay1[i]=e38329d[i]/e38329ib[i];
            if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>0 and e23901[i]>0 then
                othpay1[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e23901[i];
            if e38329i[i]=4 and e38329ib[i]>0 and e23901[i]>0 then othpay1[i]=(e38329d[i]/e38329ib[i])/e23901[i];
            if e38329i[i]=5 and e38329ib[i]>0 and e23901[i]>0 then othpay1[i]=(e38329d[i]/e38329ib[i])/(e23901[i]*4.3);
    end;

```

```

/* missing value */
if -4<e38329i[i]<0 or -4<e38329ib[i]<=0 then othpay1[i]=-3;
if e38329i[i] in (3,4,5) and -4<e23901[i]<=0 then othpay1[i]=-3;
if e38329i[i]=3 and e38329k[i]<0 then othpay1[i]=-3;
end;
end;

if e38329b[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38329d[i]<0 then
    othpay1[i]=e225001[i];
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay1[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay1[i]=e23901[i];
if e38329b[i]=2 and e38329k[i] le 0 then othpay1[i]=-3;
if e38329b[i]=2 and -4<e38329k[i]<0 then othpay1[i]=e225501[i];
if e38329b[i]=6 and e35600[i] le 0 then othpay1[i]=-3;
if e38329b[i]=6 and -4<e35600[i]<0 then othpay1[i]=e35600[i];

/*report non-hourly wage at the beginning*/
if e34402[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay1[i]=e38329d[i];
if e34402[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay1[i]=(e38329d[i]*e38329k[i])/e34402[i];
if e34402[i]>0 and e38329b[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay1[i]=e38329d[i]/e34402[i];
if e34402[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2*e34402[i]);
if e34402[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(4.3*e34402[i]);
if e34402[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay1[i]=e38329d[i]/(e35600[i]*e34402[i]);
if e34402[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2.15*e34402[i]);
if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
    if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>=0 and e34402[i]>0 then
        othpay1[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e34402[i];
    if e38329i[i]=4 and e38329ib[i]>0 and e34402[i]>0 then othpay1[i]=(e38329d[i]/e38329ib[i])/e34402[i];
    if e38329i[i]=5 and e38329ib[i]>0 and e34402[i]>0 then othpay1[i]=(e38329d[i]/e38329ib[i])/(e34402[i]*4.3);
/* missing value */
    if e38329i[i] in (3,4,5) and -4<e34402[i]<=0 then othpay1[i]=-3;
end;

if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay1[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
    othpay1[i]=e34402[i];

end;
end;

/*for commissions*/
if e212003[i]=1 then do;

/*with overtime at the beginning*/
if e38102[i] ne 1 and e38102[i] ne 3 then do;

/*report hourly wage at the beginning*/
if e23901[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay2[i]=e38329d[i];
if e23901[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay2[i]=(e38329d[i]*e38329k[i])/e23901[i];

```

```

if e23901[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay2[i]=e38329d[i]/e23901[i];
if e23901[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2*e23901[i]);
if e23901[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(4.3*e23901[i]);
if e23901[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay2[i]=e38329d[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2.15*e23901[i]);
if e38329b[i] in (9,14) then othpay2[i]=0;
if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
    if e38329g[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225302[i] in (1,2,4,5) and e225002[i]>0 and othpay2[i]>=0 then
            othpay2[i]=othpay2[i]/e225002[i]*e38329d[i];
        if e225302[i]=3 and e225002[i]>0 and e38329k[i]>0 and e225502[i]>0 and othpay2[i]>=0
            then othpay2[i]=othpay2[i]/(e225002[i]*e225502[i])*e38329d[i]*e38329k[i];
        if e225302[i]=3 and e38329k[i]<=0 then othpay2[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38329i[i]=1 and e38329ib[i]>0 then othpay2[i]=(e38329d[i]/e38329ib[i])*60;
        if e38329i[i]=2 and e38329ib[i]>0 then othpay2[i]=e38329d[i]/e38329ib[i];
        if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>0 and e23901[i]>0 then
            othpay2[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e23901[i];
        if e38329i[i]=4 and e38329ib[i]>0 and e23901[i]>0 then othpay2[i]=(e38329d[i]/e38329ib[i])/e23901[i];
        if e38329i[i]=5 and e38329ib[i]>0 and e23901[i]>0 then othpay2[i]=(e38329d[i]/e38329ib[i])/(e23901[i]*4.3);
    /* missing value */
        if -4<e38329i[i]<0 or -4<e38329ib[i]<=0 then othpay2[i]=-3;
        if e38329i[i] in (3,4,5) and -4<e23901[i]<=0 then othpay2[i]=-3;
        if e38329i[i]=3 and e38329k[i]<0 then othpay2[i]=-3;
    end;
end;

if e38329b[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38329d[i]<0 then
    othpay2[i]=e225002[i];
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay2[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay2[i]=e23901[i];
if e38329b[i]=2 and e38329k[i]<=0 then othpay2[i]=-3;
if e38329b[i]=2 and -4<e38329k[i]<0 then othpay2[i]=e225502[i];
if e38329b[i]=6 and e35600[i]<=0 then othpay2[i]=-3;
if e38329b[i]=6 and -4<e35600[i]<0 then othpay2[i]=e35600[i];

/*report nonhourly wage*/
if e34428[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay2[i]=e38329d[i];
if e34428[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay2[i]=(e38329d[i]*e38329k[i])/e34428[i];
if e34428[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay2[i]=e38329d[i]/e34428[i];
if e34428[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2*e34428[i]);
if e34428[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(4.3*e34428[i]);
if e34428[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay2[i]=e38329d[i]/(e35600[i]*e34428[i]);
if e34428[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2.15*e34428[i]);
if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
    if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>0 and e34428[i]>0 then
        othpay2[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e34428[i];
    if e38329i[i]=4 and e38329ib[i]>0 and e34428[i]>0 then othpay2[i]=(e38329d[i]/e38329ib[i])/e34428[i];
    if e38329i[i]=5 and e38329ib[i]>0 and e34428[i]>0 then othpay2[i]=(e38329d[i]/e38329ib[i])/(e34428[i]*4.3);
/* missing value */

```

Appendix 2: Employment Variable Creation

```

if e38329i[i] in (3,4,5) and -4<e34428[i]<=0 then othpay2[i]=-3;
end;

if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34428[i]=0 then othpay2[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34428[i]<0 then
    othpay2[i]=e34428[i];

end;

/*without overtime at the beginning*/
if e3800b[i]=1 then do;

/*report hourly wage at the beginning*/
if e23901[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay2[i]=e38329d[i];
if e23901[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay2[i]=(e38329d[i]*e38329k[i])/e23901[i];
if e23901[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay2[i]=e38329d[i]/e23901[i];
if e23901[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2*e23901[i]);
if e23901[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(4.3*e23901[i]);
if e23901[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay2[i]=e38329d[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2.15*e23901[i]);
if e38329b[i] in (9,14) then othpay2[i]=0;
if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
    if e38329g[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225301[i] in (1,2,4,5) and e225001[i]>0 and othpay2[i]>=0 then
            othpay2[i]=othpay2[i]/e225001[i]*e38329d[i];
        if e225301[i]=3 and e225001[i]>0 and e38329k[i]>0 and e225501[i]>0 and othpay2[i]>=0
            then othpay2[i]=othpay2[i]/(e225001[i]*e225501[i])*e38329d[i]*e38329k[i];
        if e225301[i]=3 and e38329k[i]<=0 then othpay2[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38329i[i]=1 and e38329ib[i]>0 then othpay2[i]=(e38329d[i]/e38329ib[i])*60;
        if e38329i[i]=2 and e38329ib[i]>0 then othpay2[i]=e38329d[i]/e38329ib[i];
        if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>=0 and e23901[i]>0 then
            othpay2[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e23901[i];
        if e38329i[i]=4 and e38329ib[i]>0 and e23901[i]>0 then othpay2[i]=(e38329d[i]/e38329ib[i])/e23901[i];
        if e38329i[i]=5 and e38329ib[i]>0 and e23901[i]>0 then othpay2[i]=(e38329d[i]/e38329ib[i])/(e23901[i]*4.3);
    /* missing value */
    if -4<e38329i[i]<0 or -4<e38329ib[i]<=0 then othpay2[i]=-3;
    if e38329i[i] in (3,4,5) and -4<e23901[i]<=0 then othpay2[i]=-3;
    if e38329i[i]=3 and e38329k[i]<0 then othpay2[i]=-3;
    end;
    end;

if e38329b[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38329d[i]<0 then
    othpay2[i]=e225001[i];
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay2[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay2[i]=e23901[i];
if e38329b[i]=2 and e38329k[i] le 0 then othpay2[i]=-3;
if e38329b[i]=2 and -4<e38329k[i]<0 then othpay2[i]=e225501[i];
if e38329b[i]=6 and e35600[i] le 0 then othpay2[i]=-3;
if e38329b[i]=6 and -4<e35600[i]<0 then othpay2[i]=e35600[i];

/*report nonhourly wage*/

```

Appendix 2: Employment Variable Creation

```

if e34402[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay2[i]=e38329d[i];
if e34402[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay2[i]=(e38329d[i]*e38329k[i])/e34402[i];
if e34402[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay2[i]=e38329d[i]/e34402[i];
if e34402[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2*e34402[i]);
if e34402[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(4.3*e34402[i]);
if e34402[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay2[i]=e38329d[i]/(e35600[i]*e34402[i]);
if e34402[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2.15*e34402[i]);
if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
    if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>=0 and e34402[i]>0 then
        othpay2[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e34402[i];
        if e38329i[i]=4 and e38329ib[i]>0 and e34402[i]>0 then othpay2[i]=(e38329d[i]/e38329ib[i])/e34402[i];
        if e38329i[i]=5 and e38329ib[i]>0 and e34402[i]>0 then othpay2[i]=(e38329d[i]/e38329ib[i])/(e34402[i]*4.3);
        /* missing value */
        if e38329i[i] in (3,4,5) and -4<e34402[i]<=0 then othpay2[i]=-3;
    end;
end;

if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay2[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
    othpay2[i]=e34402[i];

end;
end;

/*for bonuses*/
if e212004[i]=1 then do;
    /*with overtime at the beginning*/
    if e38102[i] ne 1 and e38102[i] ne 3 then do;
        /* report hourly wage at the beginning*/
        if e23901[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay3[i]=e38329d[i];
        if e23901[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
            othpay3[i]=(e38329d[i]*e38329k[i])/e23901[i];
        if e23901[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
            then othpay3[i]=e38329d[i]/e23901[i];
        if e23901[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2*e23901[i]);
        if e23901[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(4.3*e23901[i]);
        if e23901[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
            othpay3[i]=e38329d[i]/(e35600[i]*e23901[i]);
        if e23901[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2.15*e23901[i]);
        if e38329b[i] in (9,14) then othpay3[i]=0;
        if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
            if e38329g[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
                if e225303[i] in (1,2,4,5) and e225003[i]>0 and othpay3[i]>=0 then
                    othpay3[i]=othpay3[i]/e225003[i]*e38329d[i];
                if e225303[i]=3 and e225003[i]>0 and e38329k[i]>0 and e225503[i]>0 and othpay3[i]>=0
                    then othpay3[i]=othpay3[i]/(e225003[i]*e225503[i])*e38329d[i]*e38329k[i];
                if e225303[i]=3 and e38329k[i]<=0 then othpay3[i]=-3;
            end;
            else do; /*in all the other cases */
                if e38329i[i]=1 and e38329ib[i]>0 then othpay3[i]=(e38329d[i]/e38329ib[i])*60;
                if e38329i[i]=2 and e38329ib[i]>0 then othpay3[i]=e38329d[i]/e38329ib[i];
                if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>=0 and e23901[i]>0 then

```

Appendix 2: Employment Variable Creation

```

othpay3[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e23901[i];
if e38329i[i]=4 and e38329ib[i]>0 and e23901[i]>0 then othpay3[i]=(e38329d[i]/e38329ib[i])/e23901[i];
if e38329i[i]=5 and e38329ib[i]>0 and e23901[i]>0 then othpay3[i]=(e38329d[i]/e38329ib[i])/(e23901[i]*4.3);
/* missing value */
if -4<e38329i[i]<0 or -4<e38329ib[i]<=0 then othpay3[i]=-3;
if e38329i[i] in (3,4,5) and -4<e23901[i]<=0 then othpay3[i]=-3;
if e38329i[i]=3 and e38329k[i]<0 then othpay3[i]=-3;
end;
end;

if e38329b[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38329d[i]<0 then
    othpay3[i]=e225003[i];
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay3[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay3[i]=e23901[i];
if e38329b[i]=2 and e38329k[i] le 0 then othpay3[i]=-3;
if e38329b[i]=2 and -4<e38329k[i]<0 then othpay3[i]=e225503[i];
if e38329b[i]=6 and e35600[i] le 0 then othpay3[i]=-3;
if e38329b[i]=6 and -4<e35600[i]<0 then othpay3[i]=e35600[i];

/*report non-hourly wage*/
if e34428[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay3[i]=e38329d[i];
if e34428[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay3[i]=(e38329d[i]*e38329k[i])/e34428[i];
if e34428[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay3[i]=e38329d[i]/e34428[i];
if e34428[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2*e34428[i]);
if e34428[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(4.3*e34428[i]);
if e34428[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay3[i]=e38329d[i]/(e35600[i]*e34428[i]);
if e34428[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2.15*e34428[i]);
if e38329b[i] in (12,13) and e38329d[i]=0 then do;
    if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>=0 and e34428[i]>0 then
        othpay3[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e34428[i];
    if e38329i[i]=4 and e38329ib[i]>0 and e34428[i]>0 then othpay3[i]=(e38329d[i]/e38329ib[i])/e34428[i];
    if e38329i[i]=5 and e38329ib[i]>0 and e34428[i]>0 then othpay3[i]=(e38329d[i]/e38329ib[i])/(e34428[i]*4.3);
/* missing value */
    if e38329i[i] in (3,4,5) and -4<e34428[i]<=0 then othpay3[i]=-3;
end;

if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34428[i]=0 then othpay3[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34428[i]<0 then
    othpay3[i]=e34428[i];

end;

/*without overtime at the beginning*/
if e3800b[i]=1 then do;

/* report hourly wage at the beginning*/
if e23901[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay3[i]=e38329d[i];
if e23901[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay3[i]=(e38329d[i]*e38329k[i])/e23901[i];
if e23901[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay3[i]=e38329d[i]/e23901[i];
if e23901[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2*e23901[i]);
if e23901[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(4.3*e23901[i]);

```

```

if e23901[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay3[i]=e38329d[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2.15*e23901[i]);
if e38329b[i] in (9,14) then othpay3[i]=0;
if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
    if e38329g[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225301[i] in (1,2,4,5) and e225001[i]>0 and othpay3[i]>=0 then
            othpay3[i]=othpay3[i]/e225001[i]*e38329d[i];
        if e225301[i]=3 and e225001[i]>0 and e38329k[i]>0 and e225501[i]>0 and othpay3[i]>=0
            then othpay3[i]=othpay3[i]/(e225001[i]*e225501[i])*e38329d[i]*e38329k[i];
        if e225301[i]=3 and e38329k[i]<=0 then othpay3[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38329i[i]=1 and e38329ib[i]>0 then othpay3[i]=(e38329d[i]/e38329ib[i])*60;
        if e38329i[i]=2 and e38329ib[i]>0 then othpay3[i]=e38329d[i]/e38329ib[i];
        if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>0 and e23901[i]>0 then
            othpay3[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e23901[i];
        if e38329i[i]=4 and e38329ib[i]>0 and e23901[i]>0 then othpay3[i]=(e38329d[i]/e38329ib[i])/e23901[i];
        if e38329i[i]=5 and e38329ib[i]>0 and e23901[i]>0 then othpay3[i]=(e38329d[i]/e38329ib[i])/(e23901[i]*4.3);
        /* missing value */
        if -4<e38329i[i]<0 or -4<e38329ib[i]<=0 then othpay3[i]=-3;
        if e38329i[i] in (3,4,5) and -4<e23901[i]<=0 then othpay3[i]=-3;
        if e38329i[i]=3 and e38329k[i]<0 then othpay3[i]=-3;
    end;
end;
if e38329b[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38329d[i]<0 then
    othpay3[i]=e225001[i];
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay3[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay3[i]=e23901[i];
if e38329b[i]=2 and e38329k[i]<=0 then othpay3[i]=-3;
if e38329b[i]=2 and -4<e38329k[i]<0 then othpay3[i]=e225501[i];
if e38329b[i]=6 and e35600[i]<=0 then othpay3[i]=-3;
if e38329b[i]=6 and -4<e35600[i]<0 then othpay3[i]=e35600[i];

/*report non-hourly wage*/
if e34402[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay3[i]=e38329d[i];
if e34402[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay3[i]=(e38329d[i]*e38329k[i])/e34402[i];
if e34402[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay3[i]=e38329d[i]/e34402[i];
if e34402[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2*e34402[i]);
if e34402[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(4.3*e34402[i]);
if e34402[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay3[i]=e38329d[i]/(e35600[i]*e34402[i]);
if e34402[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2.15*e34402[i]);
if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
    if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>0 and e34402[i]>0 then
        othpay3[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e34402[i];
    if e38329i[i]=4 and e38329ib[i]>0 and e34402[i]>0 then othpay3[i]=(e38329d[i]/e38329ib[i])/e34402[i];
    if e38329i[i]=5 and e38329ib[i]>0 and e34402[i]>0 then othpay3[i]=(e38329d[i]/e38329ib[i])/(e34402[i]*4.3);
    /* missing value */
    if e38329i[i] in (3,4,5) and -4<e34402[i]<=0 then othpay3[i]=-3;
end;

if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay3[i]=-3;

```

Appendix 2: Employment Variable Creation

```

if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
    othpay3[i]=e34402[i];

end;
end;

/*for incentive pay*/

if e212005[i]=1 then do;

/*with overtime at the beginning*/
if e38102[i] ne 1 and e38102[i] ne 3 then do;

/* report hourly wage at the beginning*/
if e23901[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay4[i]=e38329d[i];
if e23901[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay4[i]=(e38329d[i]*e38329k[i])/e23901[i];
if e23901[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay4[i]=e38329d[i]/e23901[i];
if e23901[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2*e23901[i]);
if e23901[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(4.3*e23901[i]);
if e23901[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay4[i]=e38329d[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2.15*e23901[i]);
if e38329b[i] in (9,14) then othpay4[i]=0;
if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
    if e38329g[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225304[i] in (1,2,4,5) and e225004[i]>0 and othpay4[i]>=0 then
            othpay4[i]=othpay4[i]/e225004[i]*e38329d[i];
        if e225304[i]=3 and e225004[i]>0 and e38329k[i]>0 and e225504[i]>0 and othpay4[i]>=0
            then othpay4[i]=othpay4[i]/(e225004[i]*e225504[i])*e38329d[i]*e38329k[i];
        if e225304[i]=3 and e38329k[i]<=0 then othpay4[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38329i[i]=1 and e38329ib[i]>0 then othpay4[i]=(e38329d[i]/e38329ib[i])*60;
        if e38329i[i]=2 and e38329ib[i]>0 then othpay4[i]=e38329d[i]/e38329ib[i];
        if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>=0 and e23901[i]>0 then
            othpay4[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e23901[i];
        if e38329i[i]=4 and e38329ib[i]>0 and e23901[i]>0 then othpay4[i]=(e38329d[i]/e38329ib[i])/e23901[i];
        if e38329i[i]=5 and e38329ib[i]>0 and e23901[i]>0 then othpay4[i]=(e38329d[i]/e38329ib[i])/(e23901[i]*4.3);
    /* missing value */
        if -4<e38329i[i]<0 or -4<e38329ib[i]<=0 then othpay4[i]=-3;
        if e38329i[i] in (3,4,5) and -4<e23901[i]<=0 then othpay4[i]=-3;
        if e38329i[i]=3 and e38329k[i]<0 then othpay4[i]=-3;
    end;
    end;

if e38329b[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38329d[i]<0 then
    othpay4[i]=e225004[i];
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay4[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay4[i]=e23901[i];
if e38329b[i]=2 and e38329k[i] le 0 then othpay4[i]=-3;
if e38329b[i]=2 and -4<e38329k[i]<0 then othpay4[i]=e225504[i];
if e38329b[i]=6 and e35600[i] le 0 then othpay4[i]=-3;
if e38329b[i]=6 and -4<e35600[i]<0 then othpay4[i]=e35600[i];

```

```

/* report nonhourly wage*/
if e34428[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay4[i]=e38329d[i];
if e34428[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay4[i]=(e38329d[i]*e38329k[i])/e34428[i];
if e34428[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay4[i]=e38329d[i]/e34428[i];
if e34428[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2*e34428[i]);
if e34428[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(4.3*e34428[i]);
if e34428[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay4[i]=e38329d[i]/(e35600[i]*e34428[i]);
if e34428[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2.15*e34428[i]);
if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
    if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>=0 and e34428[i]>0 then
        othpay4[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e34428[i];
        if e38329i[i]=4 and e38329ib[i]>0 and e34428[i]>0 then othpay4[i]=(e38329d[i]/e38329ib[i])/e34428[i];
        if e38329i[i]=5 and e38329ib[i]>0 and e34428[i]>0 then othpay4[i]=(e38329d[i]/e38329ib[i])/(e34428[i]*4.3);
    /* missing value */
    if e38329i[i] in (3,4,5) and -4<e34428[i]<=0 then othpay4[i]=-3;
end;

if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34428[i]=0 then othpay4[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34428[i]<0 then
    othpay4[i]=e34428[i];

end;

/*without overtime at the beginning*/
if e3800b[i]=1 then do;

/* report hourly wage at the beginning*/
if e23901[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay4[i]=e38329d[i];
if e23901[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay4[i]=(e38329d[i]*e38329k[i])/e23901[i];
if e23901[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay4[i]=e38329d[i]/e23901[i];
if e23901[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2*e23901[i]);
if e23901[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(4.3*e23901[i]);
if e23901[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay4[i]=e38329d[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2.15*e23901[i]);
if e38329b[i] in (9,14) then othpay4[i]=0;
if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
    if e38329g[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225301[i] in (1,2,4,5) and e225001[i]>0 and othpay4[i]>=0 then
            othpay4[i]=othpay4[i]/e225001[i]*e38329d[i];
        if e225301[i]=3 and e225001[i]>0 and e38329k[i]>0 and e225501[i]>0 and othpay4[i]>=0
            then othpay4[i]=othpay4[i]/(e225001[i]*e225501[i])*e38329d[i]*e38329k[i];
        if e225301[i]=3 and e38329k[i]<=0 then othpay4[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38329i[i]=1 and e38329ib[i]>0 then othpay4[i]=(e38329d[i]/e38329ib[i])*60;
        if e38329i[i]=2 and e38329ib[i]>0 then othpay4[i]=e38329d[i]/e38329ib[i];
        if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>=0 and e23901[i]>0 then
            othpay4[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e23901[i];
        if e38329i[i]=4 and e38329ib[i]>0 and e23901[i]>0 then othpay4[i]=(e38329d[i]/e38329ib[i])/e23901[i];
        if e38329i[i]=5 and e38329ib[i]>0 and e23901[i]>0 then othpay4[i]=(e38329d[i]/e38329ib[i])/(e23901[i]*4.3);
    /* missing value */

```

Appendix 2: Employment Variable Creation

```

if -4<e38329i[i]<0 or -4<e38329ib[i]<=0 then othpay4[i]=-3;
if e38329i[i] in (3,4,5) and -4<e23901[i]<=0 then othpay4[i]=-3;
if e38329i[i]=3 and e38329k[i]<0 then othpay4[i]=-3;
end;
end;

if e38329b[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38329d[i]<0 then
    othpay4[i]=e225001[i];
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay4[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay4[i]=e23901[i];
if e38329b[i]=2 and e38329k[i] le 0 then othpay4[i]=-3;
if e38329b[i]=2 and -4<e38329k[i]<0 then othpay4[i]=e225501[i];
if e38329b[i]=6 and e35600[i] le 0 then othpay4[i]=-3;
if e38329b[i]=6 and -4<e35600[i]<0 then othpay4[i]=e35600[i];

/* report nonhourly wage*/
if e34402[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay4[i]=e38329d[i];
if e34402[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay4[i]=(e38329d[i]*e38329k[i])/e34402[i];
if e34402[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay4[i]=e38329d[i]/e34402[i];
if e34402[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2*e34402[i]);
if e34402[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(4.3*e34402[i]);
if e34402[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay4[i]=e38329d[i]/(e35600[i]*e34402[i]);
if e34402[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2.15*e34402[i]);
if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
    if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>=0 and e34402[i]>0 then
        othpay4[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e34402[i];
    if e38329i[i]=4 and e38329ib[i]>0 and e34402[i]>0 then othpay4[i]=(e38329d[i]/e38329ib[i])/e34402[i];
    if e38329i[i]=5 and e38329ib[i]>0 and e34402[i]>0 then othpay4[i]=(e38329d[i]/e38329ib[i])/(e34402[i]*4.3);
/* missing value */
    if e38329i[i] in (3,4,5) and -4<e34402[i]<=0 then othpay4[i]=-3;
end;

if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay4[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
    othpay4[i]=e34402[i];
end;
end;

/*for others*/
if e212006[i]=1 then do;

/*with overtime at the beginning*/
if e38102[i] ne 1 and e38102[i] ne 3 then do;

/* report hourly wage at the beginning*/
if e23901[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay5[i]=e38329d[i];
if e23901[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay5[i]=(e38329d[i]*e38329k[i])/e23901[i];
if e23901[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay5[i]=e38329d[i]/e23901[i];
if e23901[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2*e23901[i]);

```

```

if e23901[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(4.3*e23901[i]);
if e23901[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay5[i]=e38329d[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2.15*e23901[i]);
if e38329b[i] in (9,14) then othpay5[i]=0;
if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
    if e38329g[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225305[i] in (1,2,4,5) and e225005[i]>0 and othpay5[i]>=0 then
            othpay5[i]=othpay5[i]/e225005[i]*e38329d[i];
        if e225305[i]=3 and e225005[i]>0 and e38329k[i]>0 and e225505[i]>0 and othpay5[i]>=0
            then othpay5[i]=othpay5[i]/(e225005[i]*e225505[i])*e38329d[i]*e38329k[i];
        if e225305[i]=3 and e38329k[i]<=0 then othpay5[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38329i[i]=1 and e38329ib[i]>0 then othpay5[i]=(e38329d[i]/e38329ib[i])*60;
        if e38329i[i]=2 and e38329ib[i]>0 then othpay5[i]=e38329d[i]/e38329ib[i];
        if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>0 and e23901[i]>0 then
            othpay5[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e23901[i];
        if e38329i[i]=4 and e38329ib[i]>0 and e23901[i]>0 then othpay5[i]=(e38329d[i]/e38329ib[i])/e23901[i];
        if e38329i[i]=5 and e38329ib[i]>0 and e23901[i]>0 then othpay5[i]=(e38329d[i]/e38329ib[i])/(e23901[i]*4.3);
    /* missing value */
        if -4<e38329i[i]<0 or -4<e38329ib[i]<=0 then othpay5[i]=-3;
        if e38329i[i] in (3,4,5) and -4<e23901[i]<=0 then othpay5[i]=-3;
        if e38329i[i]=3 and e38329k[i]<0 then othpay5[i]=-3;
    end;
end;
if e38329b[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38329d[i]<0 then
    othpay5[i]=e225005[i];
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay5[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay5[i]=e23901[i];
if e38329b[i]=2 and e38329k[i]<0 then othpay5[i]=-3;
if e38329b[i]=2 and -4<e38329k[i]<0 then othpay5[i]=e225505[i];
if e38329b[i]=6 and e35600[i]<0 then othpay5[i]=-3;
if e38329b[i]=6 and -4<e35600[i]<0 then othpay5[i]=e35600[i];

/*report non-hourly wage*/
if e34428[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay5[i]=e38329d[i];
if e34428[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay5[i]=(e38329d[i]*e38329k[i])/e34428[i];
if e34428[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay5[i]=e38329d[i]/e34428[i];
if e34428[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2*e34428[i]);
if e34428[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(4.3*e34428[i]);
if e34428[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay5[i]=e38329d[i]/(e35600[i]*e34428[i]);
if e34428[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2.15*e34428[i]);
if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
    if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>0 and e34428[i]>0 then
        othpay5[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e34428[i];
    if e38329i[i]=4 and e38329ib[i]>0 and e34428[i]>0 then othpay5[i]=(e38329d[i]/e38329ib[i])/e34428[i];
    if e38329i[i]=5 and e38329ib[i]>0 and e34428[i]>0 then othpay5[i]=(e38329d[i]/e38329ib[i])/(e34428[i]*4.3);
    /* missing value */
    if e38329i[i] in (3,4,5) and -4<e34428[i]<=0 then othpay5[i]=-3;
end;

```

Appendix 2: Employment Variable Creation

```

if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34428[i]=0 then othpay5[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34428[i]<0 then
    othpay5[i]=e34428[i];

end;

/*without overtime at the beginning*/
if e3800b[i]=1 then do;

/* report hourly wage at the beginning*/
if e23901[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay5[i]=e38329d[i];
if e23901[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay5[i]=(e38329d[i]*e38329k[i])/e23901[i];
if e23901[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay5[i]=e38329d[i]/e23901[i];
if e23901[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2*e23901[i]);
if e23901[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(4.3*e23901[i]);
if e23901[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay5[i]=e38329d[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2.15*e23901[i]);
if e38329b[i] in (9,14) then othpay5[i]=0;
if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
    if e38329g[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225301[i] in (1,2,4,5) and e225001[i]>0 and othpay5[i]>=0 then
            othpay5[i]=othpay5[i]/e225001[i]*e38329d[i];
        if e225301[i]=3 and e225001[i]>0 and e38329k[i]>0 and e225501[i]>0 and othpay5[i]>=0
            then othpay5[i]=othpay5[i]/(e225001[i]*e225501[i])*e38329d[i]*e38329k[i];
        if e225301[i]=3 and e38329k[i]<=0 then othpay5[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38329i[i]=1 and e38329ib[i]>0 then othpay5[i]=(e38329d[i]/e38329ib[i])*60;
        if e38329i[i]=2 and e38329ib[i]>0 then othpay5[i]=e38329d[i]/e38329ib[i];
        if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>0 and e23901[i]>0 then
            othpay5[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e23901[i];
        if e38329i[i]=4 and e38329ib[i]>0 and e23901[i]>0 then othpay5[i]=(e38329d[i]/e38329ib[i])/e23901[i];
        if e38329i[i]=5 and e38329ib[i]>0 and e23901[i]>0 then othpay5[i]=(e38329d[i]/e38329ib[i])/(e23901[i]*4.3);
    /* missing value */
        if -4<e38329i[i]<0 or -4<e38329ib[i]<=0 then othpay5[i]=-3;
        if e38329i[i] in (3,4,5) and -4<e23901[i]<=0 then othpay5[i]=-3;
        if e38329i[i]=3 and e38329k[i]<0 then othpay5[i]=-3;
    end;
end;

if e38329b[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38329d[i]<0 then
    othpay5[i]=e225001[i];
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay5[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay5[i]=e23901[i];
if e38329b[i]=2 and e38329k[i] le 0 then othpay5[i]=-3;
if e38329b[i]=2 and -4<e38329k[i]<0 then othpay5[i]=e225501[i];
if e38329b[i]=6 and e35600[i] le 0 then othpay5[i]=-3;
if e38329b[i]=6 and -4<e35600[i]<0 then othpay5[i]=e35600[i];

/*report non-hourly wage*/
if e34402[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay5[i]=e38329d[i];
if e34402[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay5[i]=(e38329d[i]*e38329k[i])/e34402[i];

```

```

if e34402[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay5[i]=e38329d[i]/e34402[i];
if e34402[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2*e34402[i]);
if e34402[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(4.3*e34402[i]);
if e34402[i]>0 and e38329b[i]=6 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(52*e34402[i]);
if e34402[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2.15*e34402[i]);
if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
    if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>=0 and e34402[i]>0 then
        othpay5[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e34402[i];
    if e38329i[i]=4 and e38329ib[i]>0 and e34402[i]>0 then othpay5[i]=(e38329d[i]/e38329ib[i])/e34402[i];
    if e38329i[i]=5 and e38329ib[i]>0 and e34402[i]>0 then othpay5[i]=(e38329d[i]/e38329ib[i])/(e34402[i]*4.3);
    /* missing value */
    if e38329i[i] in (3,4,5) and -4<e34402[i]<=0 then othpay5[i]=-3;
end;

if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay5[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
    othpay5[i]=e34402[i];

end;
end;

/* case iv. one compensation at the beginning, diff no. of hours. ***/
if e38329[i]=1 and e20700[i]=1 then do;

/* for tips*/

if e212002[i]=1 then do;
    /* without overtime at the beginning*/
    if e3800b[i]=0 then do;
        if e3800ff[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay1[i]=e38329d[i];
        if e3800ff[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
            othpay1[i]=(e38329d[i]*e38329k[i])/e3800ff[i];
        if e3800ff[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
            then othpay1[i]=e38329d[i]/e3800ff[i];
        if e3800ff[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2*e3800ff[i]);
        if e3800ff[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(4.3*e3800ff[i]);
        if e3800ff[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
            othpay1[i]=e38329d[i]/(e35600[i]*e3800ff[i]);
        if e3800ff[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2.15*e3800ff[i]);

        if e38329b[i] in (9,14) then othpay1[i]=0;
        if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
            if e38329g[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
                if e225301[i] in (1,2,4,5) and e225001[i]>0 and othpay1[i]>=0 then
                    othpay1[i]=othpay1[i]/e225001[i]*e38329d[i];
                if e225301[i]=3 and e225001[i]>0 and e38329k[i]>0 and e225501[i]>0 and othpay1[i]>=0
                    then othpay1[i]=othpay1[i]/(e225001[i]*e225501[i])*e38329d[i]*e38329k[i];
                if e225301[i]=3 and e38329k[i]<=0 then othpay1[i]=-3;
            end;
            else do; /*in all the other cases */
                if e38329i[i]=1 and e38329ib[i]>0 then othpay1[i]=(e38329d[i]/e38329ib[i])*60;
                if e38329i[i]=2 and e38329ib[i]>0 then othpay1[i]=e38329d[i]/e38329ib[i];
                if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>=0 and e3800ff[i]>0 then
                    othpay1[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e3800ff[i];

```

Appendix 2: Employment Variable Creation

```

if e38329i[i]=4 and e38329ib[i]>0 and e3800ff[i]>0 then othpay1[i]=(e38329d[i]/e38329ib[i])/e3800ff[i];
if e38329i[i]=5 and e38329ib[i]>0 and e3800ff[i]>0 then othpay1[i]=(e38329d[i]/e38329ib[i])/(e3800ff[i]*4.3);
/* missing value */
if -4<e38329i[i]<0 or -4<e38329ib[i]<=0 then othpay1[i]=-3;
if e38329i[i] in (3,4,5) and -4<e3800ff[i]<=0 then othpay1[i]=-3;
if e38329i[i]=3 and e38329k[i]<0 then othpay1[i]=-3;
end;
end;

if e38329b[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38329d[i]<0 then
    othpay1[i]=e225001[i];
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e3800ff[i]=0 then othpay1[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e3800ff[i]<0 then
    othpay1[i]=e3800ff[i];
if e38329b[i]=2 and e38329k[i] le 0 then othpay1[i]=-3;
if e38329b[i]=2 and -4<e38329k[i]<0 then othpay1[i]=e225501[i];
if e38329b[i]=6 and e35600[i] le 0 then othpay1[i]=-3;
if e38329b[i]=6 and -4<e35600[i]<0 then othpay1[i]=e35600[i];

end;

/* with overtime at the beginning*/
if e38102[i]=1 or e38102[i]=3 then do;
if e38103[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay1[i]=e38329d[i];
if e38103[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay1[i]=(e38329d[i]*e38329k[i])/e38103[i];
if e38103[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay1[i]=e38329d[i]/e38103[i];
if e38103[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2*e38103[i]);
if e38103[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(4.3*e38103[i]);
if e38103[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay1[i]=e38329d[i]/(e35600[i]*e38103[i]);
if e38103[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay1[i]=e38329d[i]/(2.15*e38103[i]);
if e38103[i]>0 and e38329b[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then othpf1[i]=othpf1[i]+1;
if e38329b[i] in (9,14) then othpay1[i]=0;
if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
    if e38329g[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225301[i] in (1,2,4,5) and e225001[i]>0 and othpay1[i]>=0 then
            othpay1[i]=othpay1[i]/e225001[i]*e38329d[i];
        if e225301[i]=3 and e225001[i]>0 and e38329k[i]>0 and e225501[i]>0 and othpay1[i]>=0
            then othpay1[i]=othpay1[i]/(e225001[i]*e225501[i])*e38329d[i]*e38329k[i];
        if e225301[i]=3 and e38329k[i]<=0 then othpay1[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38329i[i]=1 and e38329ib[i]>0 then othpay1[i]=(e38329d[i]/e38329ib[i])*60;
        if e38329i[i]=2 and e38329ib[i]>0 then othpay1[i]=e38329d[i]/e38329ib[i];
        if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>=0 and e38103[i]>0 then
            othpay1[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e38103[i];
        if e38329i[i]=4 and e38329ib[i]>0 and e38103[i]>0 then othpay1[i]=(e38329d[i]/e38329ib[i])/e38103[i];
        if e38329i[i]=5 and e38329ib[i]>0 and e38103[i]>0 then othpay1[i]=(e38329d[i]/e38329ib[i])/(e38103[i]*4.3);
    /* missing value */
        if -4<e38329i[i]<0 or -4<e38329ib[i]<=0 then othpay1[i]=-3;
        if e38329i[i] in (3,4,5) and -4<e38103[i]<=0 then othpay1[i]=-3;
        if e38329i[i]=3 and e38329k[i]<0 then othpay1[i]=-3;
    end;
end;

```

```

if e38329b[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38329d[i]<0 then
    othpay1[i]=e225001[i];
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e38103[i]=0 then othpay1[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38103[i]<0 then
    othpay1[i]=e38103[i];
if e38329b[i]=2 and e38329k[i] le 0 then othpay1[i]=-3;
if e38329b[i]=2 and -4<e38329k[i]<0 then othpay1[i]=e225501[i];
if e38329b[i]=6 and e35600[i] le 0 then othpay1[i]=-3;
if e38329b[i]=6 and -4<e35600[i]<0 then othpay1[i]=e35600[i];

end;
end;

/*for commissions*/

if e212003[i]=1 then do;
/*without overtime at the beginning*/
if e3800ff[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay2[i]=e38329d[i];
if e3800ff[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay2[i]=(e38329d[i]*e38329k[i])/e3800ff[i];
if e3800ff[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay2[i]=e38329d[i]/e3800ff[i];
if e3800ff[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2*e3800ff[i]);
if e3800ff[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(4.3*e3800ff[i]);
if e3800ff[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay2[i]=e38329d[i]/(e35600[i]*e3800ff[i]);
if e3800ff[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2.15*e3800ff[i]);
if e38329b[i] in (9,14) then othpay2[i]=0;
if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
    if e38329g[i]=1 then do; /*if the speed info. given at the beginning doesn't change. */
        if e225302[i] in (1,2,4,5) and e225002[i]>0 and othpay2[i]>=0 then
            othpay2[i]=othpay2[i]/e225002[i]*e38329d[i];
        if e225302[i]=3 and e225002[i]>0 and e38329k[i]>0 and e225502[i]>0 and othpay2[i]>=0
            then othpay2[i]=othpay2[i]/(e225002[i]*e225502[i])*e38329d[i]*e38329k[i];
        if e225302[i]=3 and e38329k[i]<=0 then othpay2[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38329i[i]=1 and e38329ib[i]>0 then othpay2[i]=(e38329d[i]/e38329ib[i])*60;
        if e38329i[i]=2 and e38329ib[i]>0 then othpay2[i]=e38329d[i]/e38329ib[i];
        if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>=0 and e3800ff[i]>0 then
            othpay2[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e3800ff[i];
        if e38329i[i]=4 and e38329ib[i]>0 and e3800ff[i]>0 then othpay2[i]=(e38329d[i]/e38329ib[i])/e3800ff[i];
        if e38329i[i]=5 and e38329ib[i]>0 and e3800ff[i]>0 then othpay2[i]=(e38329d[i]/e38329ib[i])/(e3800ff[i]*4.3);
    /* missing value */
        if -4<e38329i[i]<0 or -4<e38329ib[i]<=0 then othpay2[i]=-3;
        if e38329i[i] in (3,4,5) and -4<e3800ff[i]<=0 then othpay2[i]=-3;
        if e38329i[i]=3 and e38329k[i]<0 then othpay2[i]=-3;
    end;
end;

if e38329b[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38329d[i]<0 then
    othpay2[i]=e225002[i];
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e3800ff[i]=0 then othpay2[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e3800ff[i]<0 then
    othpay2[i]=e3800ff[i];
if e38329b[i]=2 and e38329k[i] le 0 then othpay2[i]=-3;
if e38329b[i]=2 and -4<e38329k[i]<0 then othpay2[i]=e225502[i];

```

Appendix 2: Employment Variable Creation

```
if e38329b[i]=6 and e35600[i] le 0 then othpay2[i]=-3;
if e38329b[i]=6 and -4<e35600[i]<0 then othpay2[i]=e35600[i];

/* with overtime at the beginning*/
if e38102[i]=1 or e38102[i]=3 then do;

if e38103[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay2[i]=e38329d[i];
if e38103[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay2[i]=(e38329d[i]*e38329k[i])/e38103[i];
if e38103[i]>0 and e38329b[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e38329d[i]>=0
    then othpay2[i]=e38329d[i]/e38103[i];
if e38103[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2*e38103[i]);
if e38103[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(4.3*e38103[i]);
if e38103[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay2[i]=e38329d[i]/(e35600[i]*e38103[i]);
if e38103[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay2[i]=e38329d[i]/(2.15*e38103[i]);
if e38103[i]>0 and e38329b[i] in (9, 7, 12, 13, 14, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) then othpf2[i]=othpf2[i]+1;
if e38329b[i] in (9, 14) then othpay2[i]=0;
if e38329b[i] in (12, 13) and e38329d[i]>=0 then do;
    if e38329g[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225302[i] in (1, 2, 4, 5) and e225002[i]>0 and othpay2[i]>=0 then
            othpay2[i]=othpay2[i]/e225002[i]*e38329d[i];
        if e225302[i]=3 and e225002[i]>0 and e38329k[i]>0 and e225502[i]>0 and othpay2[i]>=0
            then othpay2[i]=othpay2[i]/(e225002[i]*e225502[i])*e38329d[i]*e38329k[i];
        if e225302[i]=3 and e38329k[i]<=0 then othpay2[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38329i[i]=1 and e38329ib[i]>0 then othpay2[i]=(e38329d[i]/e38329ib[i])*60;
        if e38329i[i]=2 and e38329ib[i]>0 then othpay2[i]=e38329d[i]/e38329ib[i];
        if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>=0 and e38103[i]>0 then
            othpay2[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e38103[i];
        if e38329i[i]=4 and e38329ib[i]>0 and e38103[i]>0 then othpay2[i]=(e38329d[i]/e38329ib[i])/e38103[i];
        if e38329i[i]=5 and e38329ib[i]>0 and e38103[i]>0 then othpay2[i]=(e38329d[i]/e38329ib[i])/(e38103[i]*4.3);
    /* missing value */
        if -4<e38329i[i]<0 or -4<e38329ib[i]<=0 then othpay2[i]=-3;
        if e38329i[i] in (3, 4, 5) and -4<e38103[i]<=0 then othpay2[i]=-3;
        if e38329i[i]=3 and e38329k[i]<0 then othpay2[i]=-3;
    end;
end;

if e38329b[i] in (1, 2, 3, 4, 5, 6, 7, 8, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and -4<e38329d[i]<0 then
    othpay2[i]=e225002[i];
if e38329b[i] in (2, 3, 4, 5, 6, 7, 8, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e38103[i]=0 then othpay2[i]=-3;
if e38329b[i] in (2, 3, 4, 5, 6, 7, 8, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and -4<e38103[i]<0 then
    othpay2[i]=e38103[i];
if e38329b[i]=2 and e38329k[i] le 0 then othpay2[i]=-3;
if e38329b[i]=2 and -4<e38329k[i]<0 then othpay2[i]=e225502[i];
if e38329b[i]=6 and e35600[i] le 0 then othpay2[i]=-3;
if e38329b[i]=6 and -4<e35600[i]<0 then othpay2[i]=e35600[i];

end;
end;

/*for bonuses*/

if e212004[i]=1 then do;
/* without overtime at the beginning*/
```

```

if e3800ff[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay3[i]=e38329d[i];
if e3800ff[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay3[i]=(e38329d[i]*e38329k[i])/e3800ff[i];
if e3800ff[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay3[i]=e38329d[i]/e3800ff[i];
if e3800ff[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2*e3800ff[i]);
if e3800ff[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(4.3*e3800ff[i]);
if e3800ff[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay3[i]=e38329d[i]/(e35600[i]*e3800ff[i]);
if e3800ff[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2.15*e3800ff[i]);
if e38329b[i] in (9,14) then othpay3[i]=0;
if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
    if e38329g[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225303[i] in (1,2,4,5) and e225003[i]>0 and othpay3[i]>=0 then
            othpay3[i]=othpay3[i]/e225003[i]*e38329d[i];
        if e225303[i]=3 and e225003[i]>0 and e38329k[i]>0 and e225503[i]>0 and othpay3[i]>=0
            then othpay3[i]=othpay3[i]/(e225003[i]*e225503[i])*e38329d[i]*e38329k[i];
        if e225303[i]=3 and e38329k[i]<=0 then othpay3[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38329i[i]=1 and e38329ib[i]>0 then othpay3[i]=(e38329d[i]/e38329ib[i])*60;
        if e38329i[i]=2 and e38329ib[i]>0 then othpay3[i]=e38329d[i]/e38329ib[i];
        if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>0 and e3800ff[i]>0 then
            othpay3[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e3800ff[i];
        if e38329i[i]=4 and e38329ib[i]>0 and e3800ff[i]>0 then othpay3[i]=(e38329d[i]/e38329ib[i])/e3800ff[i];
        if e38329i[i]=5 and e38329ib[i]>0 and e3800ff[i]>0 then othpay3[i]=(e38329d[i]/e38329ib[i])/(e3800ff[i]*4.3);
    /* missing value */
        if -4<e38329i[i]<0 or -4<e38329ib[i]<=0 then othpay3[i]=-3;
        if e38329i[i] in (3,4,5) and -4<e3800ff[i]<=0 then othpay3[i]=-3;
        if e38329i[i]=3 and e38329k[i]<0 then othpay3[i]=-3;
    end;
end;

if e38329b[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38329d[i]<0 then
    othpay3[i]=e225003[i];
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e3800ff[i]=0 then othpay3[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e3800ff[i]<0 then
    othpay3[i]=e3800ff[i];
if e38329b[i]=2 and e38329k[i] le 0 then othpay3[i]=-3;
if e38329b[i]=2 and -4<e38329k[i]<0 then othpay3[i]=e225503[i];
if e38329b[i]=6 and e35600[i] le 0 then othpay3[i]=-3;
if e38329b[i]=6 and -4<e35600[i]<0 then othpay3[i]=e35600[i];

/*with overtime at the beginning*/
if e38102[i]=1 or e38102[i]=3 then do;
if e38103[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay3[i]=e38329d[i];
if e38103[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay3[i]=(e38329d[i]*e38329k[i])/e38103[i];
if e38103[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay3[i]=e38329d[i]/e38103[i];
if e38103[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2*e38103[i]);
if e38103[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(4.3*e38103[i]);
if e38103[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay3[i]=e38329d[i]/(e35600[i]*e38103[i]);
if e38103[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay3[i]=e38329d[i]/(2.15*e38103[i]);
if e38329b[i] in (9,14) then othpay3[i]=0;
if e38329b[i] in (12,13) and e38329d[i]>=0 then do;

```

```

if e38329g[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
  if e225303[i] in (1,2,4,5) and e225003[i]>0 and othpay3[i]>=0 then
    othpay3[i]=othpay3[i]/e225003[i]*e38329d[i];
  if e225303[i]=3 and e225003[i]>0 and e38329k[i]>0 and e225503[i]>0 and othpay3[i]>=0
    then othpay3[i]=othpay3[i]/(e225003[i]*e225503[i])*e38329d[i]*e38329k[i];
  if e225303[i]=3 and e38329k[i]<=0 then othpay3[i]=-3;
end;
else do; /*in all the other cases */
  if e38329i[i]=1 and e38329ib[i]>0 then othpay3[i]=(e38329d[i]/e38329ib[i])*60;
  if e38329i[i]=2 and e38329ib[i]>0 then othpay3[i]=e38329d[i]/e38329ib[i];
  if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>=0 and e38103[i]>0 then
    othpay3[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e38103[i];
  if e38329i[i]=4 and e38329ib[i]>0 and e38103[i]>0 then othpay3[i]=(e38329d[i]/e38329ib[i])/e38103[i];
  if e38329i[i]=5 and e38329ib[i]>0 and e38103[i]>0 then othpay3[i]=(e38329d[i]/e38329ib[i])/(e38103[i]*4.3);
/* missing value */
  if -4<e38329i[i]<0 or -4<e38329ib[i]<=0 then othpay3[i]=-3;
  if e38329i[i] in (3,4,5) and -4<e38103[i]<=0 then othpay3[i]=-3;
  if e38329i[i]=3 and e38329k[i]<0 then othpay3[i]=-3;
end;
end;

if e38329b[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38329d[i]<0 then
  othpay3[i]=e225003[i];
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e38103[i]=0 then othpay3[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38103[i]<0 then
  othpay3[i]=e38103[i];
if e38329b[i]=2 and e38329k[i] le 0 then othpay3[i]=-3;
if e38329b[i]=2 and -4<e38329k[i]<0 then othpay3[i]=e225503[i];
if e38329b[i]=6 and e35600[i] le 0 then othpay3[i]=-3;
if e38329b[i]=6 and -4<e35600[i]<0 then othpay3[i]=e35600[i];

end;
end;

/*for incentive pay*/

if e212005[i]=1 then do;
  /*without overtime at the beginning*/
  if e3800ff[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay4[i]=e38329d[i];
  if e3800ff[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay4[i]=(e38329d[i]*e38329k[i])/e3800ff[i];
  if e3800ff[i]>0 and e38329b[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay4[i]=e38329d[i]/e3800ff[i];
  if e3800ff[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2*e3800ff[i]);
  if e3800ff[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(4.3*e3800ff[i]);
  if e3800ff[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay4[i]=e38329d[i]/(e35600[i]*e3800ff[i]);
  if e3800ff[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2.15*e3800ff[i]);
  if e38329b[i] in (9,14) then othpay4[i]=0;
  if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
    if e38329g[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
      if e225304[i] in (1,2,4,5) and e225004[i]>0 and othpay4[i]>=0 then
        othpay4[i]=othpay4[i]/e225004[i]*e38329d[i];
      if e225304[i]=3 and e225004[i]>0 and e38329k[i]>0 and e225504[i]>0 and othpay4[i]>=0
        then othpay4[i]=othpay4[i]/(e225004[i]*e225504[i])*e38329d[i]*e38329k[i];
      if e225304[i]=3 and e38329k[i]<=0 then othpay4[i]=-3;
    end;
  end;

```

```

else do; /*in all the other cases */
if e38329i[i]=1 and e38329ib[i]>0 then othpay4[i]=(e38329d[i]/e38329ib[i])*60;
if e38329i[i]=2 and e38329ib[i]>0 then othpay4[i]=e38329d[i]/e38329ib[i];
if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>=0 and e3800ff[i]>0 then
    othpay4[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e3800ff[i];
if e38329i[i]=4 and e38329ib[i]>0 and e3800ff[i]>0 then othpay4[i]=(e38329d[i]/e38329ib[i])/e3800ff[i];
if e38329i[i]=5 and e38329ib[i]>0 and e3800ff[i]>0 then othpay4[i]=(e38329d[i]/e38329ib[i])/(e3800ff[i]*4.3);
/* missing value */
if -4<e38329i[i]<0 or -4<e38329ib[i]<=0 then othpay4[i]=-3;
if e38329i[i] in (3,4,5) and -4<e3800ff[i]<=0 then othpay4[i]=-3;
if e38329i[i]=3 and e38329k[i]<0 then othpay4[i]=-3;
end;
end;

if e38329b[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38329d[i]<0 then
    othpay4[i]=e225004[i];
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e3800ff[i]=0 then othpay4[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e3800ff[i]<0 then
    othpay4[i]=e3800ff[i];
if e38329b[i]=2 and e38329k[i] le 0 then othpay4[i]=-3;
if e38329b[i]=2 and -4<e38329k[i]<0 then othpay4[i]=e225504[i];
if e38329b[i]=6 and e35600[i] le 0 then othpay4[i]=-3;
if e38329b[i]=6 and -4<e35600[i]<0 then othpay4[i]=e35600[i];

/* with overtime at the beginning*/
if e38102[i]=1 or e38102[i]=3 then do;
if e38103[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay4[i]=e38329d[i];
if e38103[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay4[i]=(e38329d[i]*e38329k[i])/e38103[i];
if e38103[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
    then othpay4[i]=e38329d[i]/e38103[i];
if e38103[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2*e38103[i]);
if e38103[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(4.3*e38103[i]);
if e38103[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay4[i]=e38329d[i]/(e35600[i]*e38103[i]);
if e38103[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay4[i]=e38329d[i]/(2.15*e38103[i]);
if e38329b[i] in (9,14) then othpay4[i]=0;
if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
    if e38329g[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225304[i] in (1,2,4,5) and e225004[i]>0 and othpay4[i]>=0 then
            othpay4[i]=othpay4[i]/e225004[i]*e38329d[i];
        if e225304[i]=3 and e225004[i]>0 and e38329k[i]>0 and e225504[i]>0 and othpay4[i]>=0
            then othpay4[i]=othpay4[i]/(e225004[i]*e225504[i])*e38329d[i]*e38329k[i];
        if e225304[i]=3 and e38329k[i]<=0 then othpay4[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38329i[i]=1 and e38329ib[i]>0 then othpay4[i]=(e38329d[i]/e38329ib[i])*60;
        if e38329i[i]=2 and e38329ib[i]>0 then othpay4[i]=e38329d[i]/e38329ib[i];
        if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>=0 and e38103[i]>0 then
            othpay4[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e38103[i];
        if e38329i[i]=4 and e38329ib[i]>0 and e38103[i]>0 then othpay4[i]=(e38329d[i]/e38329ib[i])/e38103[i];
        if e38329i[i]=5 and e38329ib[i]>0 and e38103[i]>0 then othpay4[i]=(e38329d[i]/e38329ib[i])/(e38103[i]*4.3);
/* missing value */
        if -4<e38329i[i]<0 or -4<e38329ib[i]<=0 then othpay4[i]=-3;
        if e38329i[i] in (3,4,5) and -4<e38103[i]<=0 then othpay4[i]=-3;
        if e38329i[i]=3 and e38329k[i]<0 then othpay4[i]=-3;
    end;

```

Appendix 2: Employment Variable Creation

```

end;

if e38329b[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38329d[i]<0 then
    othpay4[i]=e225004[i];
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e38103[i]=0 then othpay4[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38103[i]<0 then
    othpay4[i]=e38103[i];
if e38329b[i]=2 and e38329k[i] le 0 then othpay4[i]=-3;
if e38329b[i]=2 and -4<e38329k[i]<0 then othpay4[i]=e225504[i];
if e38329b[i]=6 and e35600[i] le 0 then othpay4[i]=-3;
if e38329b[i]=6 and -4<e35600[i]<0 then othpay4[i]=e35600[i];

end;
end;

/*for others*/

if e212006[i]=1 then do;
    /* without overtime at the beginning*/
    if e3800ff[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay5[i]=e38329d[i];
    if e3800ff[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
        othpay5[i]=(e38329d[i]*e38329k[i])/e3800ff[i];
    if e3800ff[i]>0 and e38329b[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e38329d[i]>=0
        then othpay5[i]=e38329d[i]/e3800ff[i];
    if e3800ff[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2*e3800ff[i]);
    if e3800ff[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(4.3*e3800ff[i]);
    if e3800ff[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
        othpay5[i]=e38329d[i]/(e35600[i]*e3800ff[i]);
    if e3800ff[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2.15*e3800ff[i]);
    if e38329b[i] in (9,14) then othpay5[i]=0;
    if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
        if e38329g[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
            if e225305[i] in (1,2,4,5) and e225005[i]>0 and othpay5[i]>=0 then
                othpay5[i]=othpay5[i]/e225005[i]*e38329d[i];
            if e225305[i]=3 and e225005[i]>0 and e38329k[i]>0 and e225505[i]>0 and othpay5[i]>=0
                then othpay5[i]=othpay5[i]/(e225005[i]*e225505[i])*e38329d[i]*e38329k[i];
            if e225305[i]=3 and e38329k[i]<=0 then othpay5[i]=-3;
        end;
        else do; /*in all the other cases */
            if e38329i[i]=1 and e38329ib[i]>0 then othpay5[i]=(e38329d[i]/e38329ib[i])*60;
            if e38329i[i]=2 and e38329ib[i]>0 then othpay5[i]=e38329d[i]/e38329ib[i];
            if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>=0 and e3800ff[i]>0 then
                othpay5[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e3800ff[i];
            if e38329i[i]=4 and e38329ib[i]>0 and e3800ff[i]>0 then othpay5[i]=(e38329d[i]/e38329ib[i])/e3800ff[i];
            if e38329i[i]=5 and e38329ib[i]>0 and e3800ff[i]>0 then othpay5[i]=(e38329d[i]/e38329ib[i])/(e3800ff[i]*4.3);
        /* missing value */
            if -4<e38329i[i]<0 or -4<e38329ib[i]<=0 then othpay5[i]=-3;
            if e38329i[i] in (3,4,5) and -4<e3800ff[i]<=0 then othpay5[i]=-3;
            if e38329i[i]=3 and e38329k[i]<0 then othpay5[i]=-3;
        end;
    end;

if e38329b[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38329d[i]<0 then
    othpay5[i]=e225005[i];
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e3800ff[i]=0 then othpay5[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e3800ff[i]<0 then
    othpay5[i]=e3800ff[i];

```

```

if e38329b[i]=2 and e38329k[i] le 0 then othpay5[i]=-3;
if e38329b[i]=2 and -4<e38329k[i]<0 then othpay5[i]=e225505[i];
if e38329b[i]=6 and e35600[i] le 0 then othpay5[i]=-3;
if e38329b[i]=6 and -4<e35600[i]<0 then othpay5[i]=e35600[i];

/* with overtime at the beginning*/
if e38102[i]=1 or e38102[i]=3 then do;
if e38103[i]>0 and e38329b[i]=1 and e38329d[i]>=0 then othpay5[i]=e38329d[i];
if e38103[i]>0 and e38329b[i]=2 and e38329k[i]>0 and e38329d[i]>=0 then
    othpay5[i]=(e38329d[i]*e38329k[i])/e38103[i];
if e38103[i]>0 and e38329b[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e38329d[i]>=0
    then othpay5[i]=e38329d[i]/e38103[i];
if e38103[i]>0 and e38329b[i]=4 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2*e38103[i]);
if e38103[i]>0 and e38329b[i]=5 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(4.3*e38103[i]);
if e38103[i]>0 and e38329b[i]=6 and e38329d[i]>=0 and e35600[i]>0 then
    othpay5[i]=e38329d[i]/(e35600[i]*e38103[i]);
if e38103[i]>0 and e38329b[i]=8 and e38329d[i]>=0 then othpay5[i]=e38329d[i]/(2.15*e38103[i]);
if e38329b[i] in (9,14) then othpay5[i]=0;
if e38329b[i] in (12,13) and e38329d[i]>=0 then do;
    if e38329g[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225305[i] in (1,2,4,5) and e225005[i]>0 and othpay5[i]>=0 then
            othpay5[i]=othpay5[i]/e225005[i]*e38329d[i];
        if e225305[i]=3 and e225005[i]>0 and e38329k[i]>0 and e225505[i]>0 and othpay5[i]>=0
            then othpay5[i]=othpay5[i]/(e225005[i]*e225505[i])*e38329d[i]*e38329k[i];
        if e225305[i]=3 and e38329k[i]<=0 then othpay5[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38329i[i]=1 and e38329ib[i]>0 then othpay5[i]=(e38329d[i]/e38329ib[i])*60;
        if e38329i[i]=2 and e38329ib[i]>0 then othpay5[i]=e38329d[i]/e38329ib[i];
        if e38329i[i]=3 and e38329ib[i]>0 and e38329k[i]>=0 and e38103[i]>0 then
            othpay5[i]=(e38329d[i]/e38329ib[i])*e38329k[i]/e38103[i];
        if e38329i[i]=4 and e38329ib[i]>0 and e38103[i]>0 then othpay5[i]=(e38329d[i]/e38329ib[i])/e38103[i];
        if e38329i[i]=5 and e38329ib[i]>0 and e38103[i]>0 then othpay5[i]=(e38329d[i]/e38329ib[i])/(e38103[i]*4.3);
    /* missing value */
        if -4<e38329i[i]<0 or -4<e38329ib[i]<=0 then othpay5[i]=-3;
        if e38329i[i] in (3,4,5) and -4<e38103[i]<=0 then othpay5[i]=-3;
        if e38329i[i]=3 and e38329k[i]<0 then othpay5[i]=-3;
    end;
end;

if e38329b[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999, -1, -2) and -4<e38329d[i]<0 then
    othpay5[i]=e225005[i];
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999, -1, -2) and e38103[i]=0 then othpay5[i]=-3;
if e38329b[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999, -1, -2) and -4<e38103[i]<0 then
    othpay5[i]=e38103[i];
if e38329b[i]=2 and e38329k[i] le 0 then othpay5[i]=-3;
if e38329b[i]=2 and -4<e38329k[i]<0 then othpay5[i]=e225505[i];
if e38329b[i]=6 and e35600[i] le 0 then othpay5[i]=-3;
if e38329b[i]=6 and -4<e35600[i]<0 then othpay5[i]=e35600[i];

end;
end;
end;

/** case v. more then one compensation at the beginning, same no. of hours. ***/
if e38330[i]=1 and e20700[i]=1 then do;

```

```

/* for tips*/
/*with overtime at the beginning*/
if e38102[i]>0 and e38102[i] ne 1 and e38102[i] ne 3 then do;

/*report hourly wage at the beginning*/
if e23901[i]>0 and e384071[i]=1 and e384161[i]>=0 then othpay1[i]=e384161[i];
if e23901[i]>0 and e384071[i]=2 and e38416k1[i]>0 and e384161[i]>=0 then
    othpay1[i]=(e384161[i]*e38416k1[i])/e23901[i];
if e23901[i]>0 and e384071[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e384161[i]>=0
    then othpay1[i]=e384161[i]/e23901[i];
if e23901[i]>0 and e384071[i]=4 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2*e23901[i]);
if e23901[i]>0 and e384071[i]=5 and e384161[i]>=0 then othpay1[i]=e384161[i]/(4.3*e23901[i]);
if e23901[i]>0 and e384071[i]=6 and e384161[i]>=0 and e35600[i]>0 then
    othpay1[i]=e384161[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e384071[i]=8 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2.15*e23901[i]);
if e384071[i] in (9, 14) then othpay1[i]=0;
if e384071[i] in (12, 13) and e384161[i]>=0 then do;
    if e38416g1[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225301[i] in (1, 2, 4, 5) and e225001[i]>0 and othpay1[i]>=0 then
            othpay1[i]=othpay1[i]/e225001[i]*e384161[i];
        if e225301[i]=3 and e225001[i]>0 and e38416k1[i]>0 and e225501[i]>0 and othpay1[i]>=0
            then othpay1[i]=othpay1[i]/(e225001[i]*e225501[i])*e384161[i]*e38416k1[i];
        if e225301[i]=3 and e38416k1[i]<=0 then othpay1[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38416i1[i]=1 and e38416ib1[i]>0 then othpay1[i]=(e384161[i]/e38416ib1[i])*60;
        if e38416i1[i]=2 and e38416ib1[i]>0 then othpay1[i]=e384161[i]/e38416ib1[i];
        if e38416i1[i]=3 and e38416ib1[i]>0 and e38416k1[i]>=0 and e23901[i]>0 then
            othpay1[i]=(e384161[i]/e38416ib1[i])*e38416k1[i]/e23901[i];
        if e38416i1[i]=4 and e38416ib1[i]>0 and e23901[i]>0 then othpay1[i]=(e384161[i]/e38416ib1[i])/e23901[i];
        if e38416i1[i]=5 and e38416ib1[i]>0 and e23901[i]>0 then othpay1[i]=(e384161[i]/e38416ib1[i])/(e23901[i]*4.3);
    /* missing value */
        if -4<e38416i1[i]<0 or -4<e38416ib1[i]<=0 then othpay1[i]=-3;
        if e38416i1[i] in (3, 4, 5) and -4<e23901[i]<=0 then othpay1[i]=-3;
        if e38416i1[i]=3 and e38416k1[i]<0 then othpay1[i]=-3;
    end;
end;
else do; /*in all the other cases */
    if e384071[i] in (1, 2, 3, 4, 5, 6, 7, 8, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and -4<e384161[i]<0 then
        othpay1[i]=e225001[i];
    if e384071[i] in (2, 3, 4, 5, 6, 7, 8, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e23901[i]=0 then othpay1[i]=-3;
    if e384071[i] in (2, 3, 4, 5, 6, 7, 8, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and -4<e23901[i]<0 then
        othpay1[i]=e23901[i];
    if e384071[i]=2 and e38416k1[i]<0 then othpay1[i]=-3;
    if e384071[i]=2 and -4<e38416k1[i]<0 then othpay1[i]=e225501[i];
    if e384071[i]=6 and e35600[i]<0 then othpay1[i]=-3;
    if e384071[i]=6 and -4<e35600[i]<0 then othpay1[i]=e35600[i];

/*report non-hourly wage*/
if e34428[i]>0 and e384071[i]=1 and e384161[i]>=0 then othpay1[i]=e384161[i];
if e34428[i]>0 and e384071[i]=2 and e38416k1[i]>0 and e384161[i]>=0 then
    othpay1[i]=(e384161[i]*e38416k1[i])/e34428[i];
if e34428[i]>0 and e384071[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e384161[i]>=0
    then othpay1[i]=e384161[i]/e34428[i];
if e34428[i]>0 and e384071[i]=4 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2*e34428[i]);
if e34428[i]>0 and e384071[i]=5 and e384161[i]>=0 then othpay1[i]=e384161[i]/(4.3*e34428[i]);

```

```

if e34428[i]>0 and e384071[i]=6 and e384161[i]>=0 and e35600[i]>0 then
    othpay1[i]=e384161[i]/(e35600[i]*e34428[i]);
if e34428[i]>0 and e384071[i]=8 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2.15*e34428[i]);
if e34428[i]>0 and e384071[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then othpf1[i]=othpf1[i]+1;
if e384071[i] in (12,13) and e384161[i]>=0 then do;
    if e384161i1[i]=3 and e38416ib1[i]>0 and e38416k1[i]>=0 and e34428[i]>0 then
        othpay1[i]=(e384161[i]/e38416ib1[i])*e38416k1[i]/e34428[i];
    if e38416i1[i]=4 and e38416ib1[i]>0 and e34428[i]>0 then othpay1[i]=(e384161[i]/e38416ib1[i])/e34428[i];
    if e38416i1[i]=5 and e38416ib1[i]>0 and e34428[i]>0 then othpay1[i]=(e384161[i]/e38416ib1[i])/(e34428[i]*4.3);
    /* missing value */
    if e38416i1[i] in (3,4,5) and -4<e34428[i]<=0 then othpay1[i]=-3;
end;

if e384071[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34428[i]=0 then othpay1[i]=-3;
if e384071[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34428[i]<0 then
    othpay1[i]=e34428[i];

end;

/*without overtime at the beginning*/
if e3800b[i]=1 then do;

/*report hourly wage at the beginning*/
if e23901[i]>0 and e384071[i]=1 and e384161[i]>=0 then othpay1[i]=e384161[i];
if e23901[i]>0 and e384071[i]=2 and e38416k1[i]>0 and e384161[i]>=0 then
    othpay1[i]=(e384161[i]*e38416k1[i])/e23901[i];
if e23901[i]>0 and e384071[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384161[i]>=0
    then othpay1[i]=e384161[i]/e23901[i];
if e23901[i]>0 and e384071[i]=4 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2*e23901[i]);
if e23901[i]>0 and e384071[i]=5 and e384161[i]>=0 then othpay1[i]=e384161[i]/(4.3*e23901[i]);
if e23901[i]>0 and e384071[i]=6 and e384161[i]>=0 and e35600[i]>0 then
    othpay1[i]=e384161[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e384071[i]=8 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2.15*e23901[i]);
if e384071[i] in (9,14) then othpay1[i]=0;
if e384071[i] in (12,13) and e384161[i]>=0 then do;
    if e38416g1[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225301[i] in (1,2,4,5) and e225001[i]>0 and othpay1[i]>=0 then
            othpay1[i]=othpay1[i]/e225001[i]*e384161[i];
        if e225301[i]=3 and e225001[i]>0 and e38416k1[i]>0 and e225501[i]>0 and othpay1[i]>=0
            then othpay1[i]=othpay1[i]/(e225001[i]*e225501[i])*e384161[i]*e38416k1[i];
        if e225301[i]=3 and e38416k1[i]<=0 then othpay1[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38416i1[i]=1 and e38416ib1[i]>0 then othpay1[i]=(e384161[i]/e38416ib1[i])*60;
        if e38416i1[i]=2 and e38416ib1[i]>0 then othpay1[i]=e384161[i]/e38416ib1[i];
        if e38416i1[i]=3 and e38416ib1[i]>0 and e38416k1[i]>=0 and e23901[i]>0 then
            othpay1[i]=(e384161[i]/e38416ib1[i])*e38416k1[i]/e23901[i];
        if e38416i1[i]=4 and e38416ib1[i]>0 and e23901[i]>0 then othpay1[i]=(e384161[i]/e38416ib1[i])/e23901[i];
        if e38416i1[i]=5 and e38416ib1[i]>0 and e23901[i]>0 then othpay1[i]=(e384161[i]/e38416ib1[i])/(e23901[i]*4.3);
    /* missing value */
        if -4<e38416i1[i]<0 or -4<e38416ib1[i]<=0 then othpay1[i]=-3;
        if e38416i1[i] in (3,4,5) and -4<e23901[i]<=0 then othpay1[i]=-3;
        if e38416i1[i]=3 and e38416k1[i]<0 then othpay1[i]=-3;
    end;
end;

```

Appendix 2: Employment Variable Creation

```

if e384071[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e384161[i]<0 then
    othpay1[i]=e225001[i];
if e384071[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay1[i]=-3;
if e384071[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay1[i]=e23901[i];
if e384071[i]=2 and e38416k1[i] le 0 then othpay1[i]=-3;
if e384071[i]=2 and -4<e38416k1[i]<0 then othpay1[i]=e225501[i];
if e384071[i]=6 and e35600[i] le 0 then othpay1[i]=-3;
if e384071[i]=6 and -4<e35600[i]<0 then othpay1[i]=e35600[i];

/*report non-hourly wage*/
if e34402[i]>0 and e384071[i]=1 and e384161[i]>=0 then othpay1[i]=e384161[i];
if e34402[i]>0 and e384071[i]=2 and e38416k1[i]>0 and e384161[i]>=0 then
    othpay1[i]=(e384161[i]*e38416k1[i])/e34402[i];
if e34402[i]>0 and e384071[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384161[i]>=0
    then othpay1[i]=e384161[i]/e34402[i];
if e34402[i]>0 and e384071[i]=4 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2*e34402[i]);
if e34402[i]>0 and e384071[i]=5 and e384161[i]>=0 then othpay1[i]=e384161[i]/(4.3*e34402[i]);
if e34402[i]>0 and e384071[i]=6 and e384161[i]>=0 and e35600[i]>0 then
    othpay1[i]=e384161[i]/(e35600[i]*e34402[i]);
if e34402[i]>0 and e384071[i]=8 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2.15*e34402[i]);
if e34402[i]>0 and e384071[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then othpf1[i]=othpf1[i]+1;
if e384071[i] in (12,13) and e384161[i]>=0 then do;
    if e38416i1[i]=3 and e38416ib1[i]>0 and e38416k1[i]>=0 and e34402[i]>0 then
        othpay1[i]=(e384161[i]/e38416ib1[i])*e38416k1[i]/e34402[i];
    if e38416i1[i]=4 and e38416ib1[i]>0 and e34402[i]>0 then othpay1[i]=(e384161[i]/e38416ib1[i])/e34402[i];
    if e38416i1[i]=5 and e38416ib1[i]>0 and e34402[i]>0 then othpay1[i]=(e384161[i]/e38416ib1[i])/(e34402[i]*4.3);
/* missing value */
    if e38416i1[i] in (3,4,5) and -4<e34402[i]<=0 then othpay1[i]=-3;
end;

if e384071[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay1[i]=-3;
if e384071[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
    othpay1[i]=e34402[i];

end;

/*for commissions*/

/*with overtime at the beginning*/
if e38102[i]>0 and e38102[i] ne 1 and e38102[i] ne 3 then do;

/* report hourly wage at the beginning*/
if e23901[i]>0 and e384072[i]=1 and e384162[i]>=0 then othpay2[i]=e384162[i];
if e23901[i]>0 and e384072[i]=2 and e38416k2[i]>0 and e384162[i]>=0 then
    othpay2[i]=(e384162[i]*e38416k2[i])/e23901[i];
if e23901[i]>0 and e384072[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384162[i]>=0
    then othpay2[i]=e384162[i]/e23901[i];
if e23901[i]>0 and e384072[i]=4 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2*e23901[i]);
if e23901[i]>0 and e384072[i]=5 and e384162[i]>=0 then othpay2[i]=e384162[i]/(4.3*e23901[i]);
if e23901[i]>0 and e384072[i]=6 and e384162[i]>=0 and e35600[i]>0 then
    othpay2[i]=e384162[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e384072[i]=8 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2.15*e23901[i]);
if e384072[i] in (9,14) then othpay2[i]=0;
if e384072[i] in (12,13) and e384162[i]>=0 then do;
    if e38416g2[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */

```

```

if e225302[i] in (1,2,4,5) and e225002[i]>0 and othpay2[i]>=0 then
    othpay2[i]=othpay2[i]/e225002[i]*e384162[i];
if e225302[i]=3 and e225002[i]>0 and e38416k2[i]>0 and e225502[i]>0 and othpay2[i]>=0
    then othpay2[i]=othpay2[i]/(e225002[i]*e225502[i])*e384162[i]*e38416k2[i];
if e225302[i]=3 and e38416k2[i]<=0 then othpay2[i]=-3;
end;
else do; /*in all the other cases */
    if e38416i2[i]=1 and e38416ib2[i]>0 then othpay2[i]=(e384162[i]/e38416ib2[i])*60;
    if e38416i2[i]=2 and e38416ib2[i]>0 then othpay2[i]=e384162[i]/e38416ib2[i];
    if e38416i2[i]=3 and e38416ib2[i]>0 and e38416k2[i]>=0 and e23901[i]>0 then
        othpay2[i]=(e384162[i]/e38416ib2[i])*e38416k2[i]/e23901[i];
    if e38416i2[i]=4 and e38416ib2[i]>0 and e23901[i]>0 then othpay2[i]=(e384162[i]/e38416ib2[i])/e23901[i];
    if e38416i2[i]=5 and e38416ib2[i]>0 and e23901[i]>0 then othpay2[i]=(e384162[i]/e38416ib2[i])/(e23901[i]*4.3);
/* missing value */
    if -4<e38416i2[i]<0 or -4<e38416ib2[i]<=0 then othpay2[i]=-3;
    if e38416i2[i] in (3,4,5) and -4<e23901[i]<=0 then othpay2[i]=-3;
    if e38416i2[i]=3 and e38416k2[i]<0 then othpay2[i]=-3;
end;
end;

if e384072[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e384162[i]<0 then
    othpay2[i]=e225002[i];
if e384072[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay2[i]=-3;
if e384072[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay2[i]=e23901[i];
if e384072[i]=2 and e38416k2[i] le 0 then othpay2[i]=-3;
if e384072[i]=2 and -4<e38416k2[i]<0 then othpay2[i]=e225502[i];
if e384072[i]=6 and e35600[i] le 0 then othpay2[i]=-3;
if e384072[i]=6 and -4<e35600[i]<0 then othpay2[i]=e35600[i];

/* report non-hourly wage*/
if e34428[i]>0 and e384072[i]=1 and e384162[i]>=0 then othpay2[i]=e384162[i];
if e34428[i]>0 and e384072[i]=2 and e38416k2[i]>0 and e384162[i]>=0 then
    othpay2[i]=(e384162[i]*e38416k2[i])/e34428[i];
if e34428[i]>0 and e384072[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384162[i]>=0
    then othpay2[i]=e384162[i]/e34428[i];
if e34428[i]>0 and e384072[i]=4 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2*e34428[i]);
if e34428[i]>0 and e384072[i]=5 and e384162[i]>=0 then othpay2[i]=e384162[i]/(4.3*e34428[i]);
if e34428[i]>0 and e384072[i]=6 and e384162[i]>=0 and e35600[i]>0 then
    othpay2[i]=e384162[i]/(e35600[i]*e34428[i]);
if e34428[i]>0 and e384072[i]=8 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2.15*e34428[i]);
if e384072[i] in (12,13) and e384162[i]>=0 then do;
    if e38416i2[i]=3 and e38416ib2[i]>0 and e38416k2[i]>=0 and e34428[i]>0 then
        othpay2[i]=(e384162[i]/e38416ib2[i])*e38416k2[i]/e34428[i];
    if e38416i2[i]=4 and e38416ib2[i]>0 and e34428[i]>0 then othpay2[i]=(e384162[i]/e38416ib2[i])/e34428[i];
    if e38416i2[i]=5 and e38416ib2[i]>0 and e34428[i]>0 then othpay2[i]=(e384162[i]/e38416ib2[i])/(e34428[i]*4.3);
/* missing value */
    if e38416i2[i] in (3,4,5) and -4<e34428[i]<=0 then othpay2[i]=-3;
end;

if e384072[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34428[i]=0 then othpay2[i]=-3;
if e384072[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34428[i]<0 then
    othpay2[i]=e34428[i];
end;

/*without overtime at the beginning*/

```

```

if e3800b[i]=1 then do;
/* report hourly wage at the beginning*/
if e23901[i]>0 and e384072[i]=1 and e384162[i]>=0 then othpay2[i]=e384162[i];
if e23901[i]>0 and e384072[i]=2 and e38416k2[i]>0 and e384162[i]>=0 then
    othpay2[i]=(e384162[i]*e38416k2[i])/e23901[i];
if e23901[i]>0 and e384072[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e384162[i]>=0
    then othpay2[i]=e384162[i]/e23901[i];
if e23901[i]>0 and e384072[i]=4 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2*e23901[i]);
if e23901[i]>0 and e384072[i]=5 and e384162[i]>=0 then othpay2[i]=e384162[i]/(4.3*e23901[i]);
if e23901[i]>0 and e384072[i]=6 and e384162[i]>=0 and e35600[i]>0 then
    othpay2[i]=e384162[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e384072[i]=8 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2.15*e23901[i]);
if e384072[i] in (9,14) then othpay2[i]=0;
if e384072[i] in (12,13) and e384162[i]>=0 then do;
    if e38416g2[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225302[i] in (1,2,4,5) and e225002[i]>0 and othpay2[i]>=0 then
            othpay2[i]=othpay2[i]/e225002[i]*e384162[i];
        if e225302[i]=3 and e225002[i]>0 and e38416k2[i]>0 and e225502[i]>0 and othpay2[i]>=0
            then othpay2[i]=othpay2[i]/(e225002[i]*e225502[i])*e384162[i]*e38416k2[i];
        if e225302[i]=3 and e38416k2[i]<=0 then othpay2[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38416i2[i]=1 and e38416ib2[i]>0 then othpay2[i]=(e384162[i]/e38416ib2[i])*60;
        if e38416i2[i]=2 and e38416ib2[i]>0 then othpay2[i]=e384162[i]/e38416ib2[i];
        if e38416i2[i]=3 and e38416ib2[i]>0 and e38416k2[i]>=0 and e23901[i]>0 then
            othpay2[i]=(e384162[i]/e38416ib2[i])*e38416k2[i]/e23901[i];
        if e38416i2[i]=4 and e38416ib2[i]>0 and e23901[i]>0 then othpay2[i]=(e384162[i]/e38416ib2[i])/e23901[i];
        if e38416i2[i]=5 and e38416ib2[i]>0 and e23901[i]>0 then othpay2[i]=(e384162[i]/e38416ib2[i])/(e23901[i]*4.3);
    /* missing value */
        if -4<e38416i2[i]<0 or -4<e38416ib2[i]<=0 then othpay2[i]=-3;
        if e38416i2[i] in (3,4,5) and -4<e23901[i]<=0 then othpay2[i]=-3;
        if e38416i2[i]=3 and e38416k2[i]<0 then othpay2[i]=-3;
    end;
end;
if e384072[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999, -1, -2) and -4<e384162[i]<0 then
    othpay2[i]=e225002[i];
if e384072[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999, -1, -2) and e23901[i]=0 then othpay2[i]=-3;
if e384072[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999, -1, -2) and -4<e23901[i]<0 then
    othpay2[i]=e23901[i];
if e384072[i]=2 and e38416k2[i] le 0 then othpay2[i]=-3;
if e384072[i]=2 and -4<e38416k2[i]<0 then othpay2[i]=e225502[i];
if e384072[i]=6 and e35600[i] le 0 then othpay2[i]=-3;
if e384072[i]=6 and -4<e35600[i]<0 then othpay2[i]=e35600[i];

/* report non-hourly wage*/
if e34402[i]>0 and e384072[i]=1 and e384162[i]>=0 then othpay2[i]=e384162[i];
if e34402[i]>0 and e384072[i]=2 and e38416k2[i]>0 and e384162[i]>=0 then
    othpay2[i]=(e384162[i]*e38416k2[i])/e34402[i];
if e34402[i]>0 and e384072[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e384162[i]>=0
    then othpay2[i]=e384162[i]/e34402[i];
if e34402[i]>0 and e384072[i]=4 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2*e34402[i]);
if e34402[i]>0 and e384072[i]=5 and e384162[i]>=0 then othpay2[i]=e384162[i]/(4.3*e34402[i]);
if e34402[i]>0 and e384072[i]=6 and e384162[i]>=0 and e35600[i]>0 then
    othpay2[i]=e384162[i]/(e35600[i]*e34402[i]);
if e34402[i]>0 and e384072[i]=8 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2.15*e34402[i]);

```

```

if e34402[i]>0 and e384072[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then othpf2[i]=othpf2[i]+1;
if e384072[i] in (12,13) and e384162[i]>=0 then do;
    if e38416i2[i]=3 and e38416ib2[i]>0 and e38416k2[i]>=0 and e34402[i]>0 then
        othpay2[i]=(e384162[i]/e38416ib2[i])*e38416k2[i]/e34402[i];
    if e38416i2[i]=4 and e38416ib2[i]>0 and e34402[i]>0 then othpay2[i]=(e384162[i]/e38416ib2[i])/e34402[i];
    if e38416i2[i]=5 and e38416ib2[i]>0 and e34402[i]>0 then othpay2[i]=(e384162[i]/e38416ib2[i])/(e34402[i]*4.3);
    /* missing value */
    if e38416i2[i] in (3,4,5) and -4<e34402[i]<=0 then othpay2[i]=-3;
end;

if e384072[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay2[i]=-3;
if e384072[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
    othpay2[i]=e34402[i];

end;

/*for bonuses*/

/*with overtime at the beginning*/
if e38102[i]>0 and e38102[i] ne 1 and e38102[i] ne 3 then do;

/* report hourly wage at the beginning*/
if e23901[i]>0 and e384073[i]=1 and e384163[i]>=0 then othpay3[i]=e384163[i];
if e23901[i]>0 and e384073[i]=2 and e38416k3[i]>0 and e384163[i]>=0 then
    othpay3[i]=(e384163[i]*e38416k3[i])/e23901[i];
if e23901[i]>0 and e384073[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384163[i]>=0
    then othpay3[i]=e384163[i]/e23901[i];
if e23901[i]>0 and e384073[i]=4 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2*e23901[i]);
if e23901[i]>0 and e384073[i]=5 and e384163[i]>=0 then othpay3[i]=e384163[i]/(4.3*e23901[i]);
if e23901[i]>0 and e384073[i]=6 and e384163[i]>=0 and e35600[i]>0 then
    othpay3[i]=e384163[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e384073[i]=8 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2.15*e23901[i]);
if e384073[i] in (9,14) then othpay3[i]=0;
if e384073[i] in (12,13) and e384163[i]>=0 then do;
    if e38416g3[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225303[i] in (1,2,4,5) and e225003[i]>0 and othpay3[i]>=0 then
            othpay3[i]=othpay3[i]/e225003[i]*e384163[i];
        if e225303[i]=3 and e225003[i]>0 and e38416k3[i]>0 and e225503[i]>0 and othpay3[i]>=0
            then othpay3[i]=othpay3[i]/(e225003[i]*e225503[i])*e384163[i]*e38416k3[i];
        if e225303[i]=3 and e38416k3[i]<=0 then othpay3[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38416i3[i]=1 and e38416ib3[i]>0 then othpay3[i]=(e384163[i]/e38416ib3[i])*60;
        if e38416i3[i]=2 and e38416ib3[i]>0 then othpay3[i]=e384163[i]/e38416ib3[i];
        if e38416i3[i]=3 and e38416ib3[i]>0 and e38416k3[i]>=0 and e23901[i]>0 then
            othpay3[i]=(e384163[i]/e38416ib3[i])*e38416k3[i]/e23901[i];
        if e38416i3[i]=4 and e38416ib3[i]>0 and e23901[i]>0 then othpay3[i]=(e384163[i]/e38416ib3[i])/e23901[i];
        if e38416i3[i]=5 and e38416ib3[i]>0 and e23901[i]>0 then othpay3[i]=(e384163[i]/e38416ib3[i])/(e23901[i]*4.3);
    /* missing value */
        if -4<e38416i3[i]<0 or -4<e38416ib3[i]<=0 then othpay3[i]=-3;
        if e38416i3[i] in (3,4,5) and -4<e23901[i]<=0 then othpay3[i]=-3;
        if e38416i3[i]=3 and e38416k3[i]<0 then othpay3[i]=-3;
    end;
end;

if e384073[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e384163[i]<0 then
    othpay3[i]=e225003[i];

```

Appendix 2: Employment Variable Creation

```

if e384073[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay3[i]=-3;
if e384073[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay3[i]=e23901[i];
if e384073[i]=2 and e38416k3[i] le 0 then othpay3[i]=-3;
if e384073[i]=2 and -4<e38416k3[i]<0 then othpay3[i]=e225503[i];
if e384073[i]=6 and e35600[i] le 0 then othpay3[i]=-3;
if e384073[i]=6 and -4<e35600[i]<0 then othpay3[i]=e35600[i];

/*report non-hourly wage*/
if e34428[i]>0 and e384073[i]=1 and e384163[i]>=0 then othpay3[i]=e384163[i];
if e34428[i]>0 and e384073[i]=2 and e38416k3[i]>0 and e384163[i]>=0 then
    othpay3[i]=(e384163[i]*e38416k3[i])/e34428[i];
if e34428[i]>0 and e384073[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384163[i]>=0
    then othpay3[i]=e384163[i]/e34428[i];
if e34428[i]>0 and e384073[i]=4 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2*e34428[i]);
if e34428[i]>0 and e384073[i]=5 and e384163[i]>=0 then othpay3[i]=e384163[i]/(4.3*e34428[i]);
if e34428[i]>0 and e384073[i]=6 and e384163[i]>=0 and e35600[i]>0 then
    othpay3[i]=e384163[i]/(e35600[i]*e34428[i]);
if e34428[i]>0 and e384073[i]=8 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2.15*e34428[i]);
if e384073[i] in (12,13) and e384163[i]>=0 then do;
    if e38416i3[i]=3 and e38416ib3[i]>0 and e38416k3[i]>=0 and e34428[i]>0 then
        othpay3[i]=(e384163[i]/e38416ib3[i])*e38416k3[i]/e34428[i];
    if e38416i3[i]=4 and e38416ib3[i]>0 and e34428[i]>0 then othpay3[i]=(e384163[i]/e38416ib3[i])/e34428[i];
    if e38416i3[i]=5 and e38416ib3[i]>0 and e34428[i]>0 then othpay3[i]=(e384163[i]/e38416ib3[i])/(e34428[i]*4.3);
/* missing value */
    if e38416i3[i] in (3,4,5) and -4<e34428[i]<=0 then othpay3[i]=-3;
end;

if e384073[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34428[i]=0 then othpay3[i]=-3;
if e384073[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34428[i]<0 then
    othpay3[i]=e34428[i];

end;

/*without overtime at the beginning*/
if e3800b[i]=1 then do;

/* report hourly wage at the beginning*/
if e23901[i]>0 and e384073[i]=1 and e384163[i]>=0 then othpay3[i]=e384163[i];
if e23901[i]>0 and e384073[i]=2 and e38416k3[i]>0 and e384163[i]>=0 then
    othpay3[i]=(e384163[i]*e38416k3[i])/e23901[i];
if e23901[i]>0 and e384073[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384163[i]>=0
    then othpay3[i]=e384163[i]/e23901[i];
if e23901[i]>0 and e384073[i]=4 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2*e23901[i]);
if e23901[i]>0 and e384073[i]=5 and e384163[i]>=0 then othpay3[i]=e384163[i]/(4.3*e23901[i]);
if e23901[i]>0 and e384073[i]=6 and e384163[i]>=0 and e35600[i]>0 then
    othpay3[i]=e384163[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e384073[i]=8 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2.15*e23901[i]);
if e384073[i] in (9,14) then othpay3[i]=0;
if e384073[i] in (12,13) and e384163[i]>=0 then do;
    if e38416g3[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225303[i] in (1,2,4,5) and e225003[i]>0 and othpay3[i]>=0 then
            othpay3[i]=othpay3[i]/e225003[i]*e384163[i];
        if e225303[i]=3 and e225003[i]>0 and e38416k3[i]>0 and e225503[i]>0 and othpay3[i]>=0
            then othpay3[i]=othpay3[i]/(e225003[i]*e225503[i])*e384163[i]*e38416k3[i];
        if e225303[i]=3 and e38416k3[i]<=0 then othpay3[i]=-3;
    end;

```

```

else do; /*in all the other cases */
if e38416i3[i]=1 and e38416ib3[i]>0 then othpay3[i]=(e384163[i]/e38416ib3[i])*60;
if e38416i3[i]=2 and e38416ib3[i]>0 then othpay3[i]=e384163[i]/e38416ib3[i];
if e38416i3[i]=3 and e38416ib3[i]>0 and e38416k3[i]>=0 and e23901[i]>0 then
    othpay3[i]=(e384163[i]/e38416ib3[i])*e38416k3[i]/e23901[i];
if e38416i3[i]=4 and e38416ib3[i]>0 and e23901[i]>0 then othpay3[i]=(e384163[i]/e38416ib3[i])/e23901[i];
if e38416i3[i]=5 and e38416ib3[i]>0 and e23901[i]>0 then othpay3[i]=(e384163[i]/e38416ib3[i])/(e23901[i]*4.3);
/* missing value */
if -4<e38416i3[i]<0 or -4<e38416ib3[i]<=0 then othpay3[i]=-3;
if e38416i3[i] in (3,4,5) and -4<e23901[i]<=0 then othpay3[i]=-3;
if e38416i3[i]=3 and e38416k3[i]<0 then othpay3[i]=-3;
end;
end;

if e384073[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e384163[i]<0 then
    othpay3[i]=e225003[i];
if e384073[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay3[i]=-3;
if e384073[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay3[i]=e23901[i];
if e384073[i]=2 and e38416k3[i] le 0 then othpay3[i]=-3;
if e384073[i]=2 and -4<e38416k3[i]<0 then othpay3[i]=e225503[i];
if e384073[i]=6 and e35600[i] le 0 then othpay3[i]=-3;
if e384073[i]=6 and -4<e35600[i]<0 then othpay3[i]=e35600[i];

/*report non-hurly wage*/
if e34402[i]>0 and e384073[i]=1 and e384163[i]>=0 then othpay3[i]=e384163[i];
if e34402[i]>0 and e384073[i]=2 and e38416k3[i]>0 and e384163[i]>=0 then
    othpay3[i]=(e384163[i]*e38416k3[i])/e34402[i];
if e34402[i]>0 and e384073[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384163[i]>=0
    then othpay3[i]=e384163[i]/e34402[i];
if e34402[i]>0 and e384073[i]=4 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2*e34402[i]);
if e34402[i]>0 and e384073[i]=5 and e384163[i]>=0 then othpay3[i]=e384163[i]/(4.3*e34402[i]);
if e34402[i]>0 and e384073[i]=6 and e384163[i]>=0 and e35600[i]>0 then
    othpay3[i]=e384163[i]/(e35600[i]*e34402[i]);
if e34402[i]>0 and e384073[i]=8 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2.15*e34402[i]);
if e34402[i]>0 and e384073[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then othpf3[i]=othpf3[i]+1;
if e384073[i] in (12,13) and e384163[i]>=0 then do;
    if e38416i3[i]=3 and e38416ib3[i]>0 and e38416k3[i]>=0 and e34402[i]>0 then
        othpay3[i]=(e384163[i]/e38416ib3[i])*e38416k3[i]/e34402[i];
    if e38416i3[i]=4 and e38416ib3[i]>0 and e34402[i]>0 then othpay3[i]=(e384163[i]/e38416ib3[i])/e34402[i];
    if e38416i3[i]=5 and e38416ib3[i]>0 and e34402[i]>0 then othpay3[i]=(e384163[i]/e38416ib3[i])/(e34402[i]*4.3);
/* missing value */
    if e38416i3[i] in (3,4,5) and -4<e34402[i]<=0 then othpay3[i]=-3;
end;

if e384073[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay3[i]=-3;
if e384073[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
    othpay3[i]=e34402[i];

end;

/*for incentive pay*/

/*with overtime at the beginning*/
if e38102[i]>0 and e38102[i] ne 1 and e38102[i] ne 3 then do;

/*report hourly wage at the beginning*/

```

```

if e23901[i]>0 and e384074[i]=1 and e384164[i]>=0 then othpay4[i]=e384164[i];
if e23901[i]>0 and e384074[i]=2 and e38416k4[i]>0 and e384164[i]>=0 then
    othpay4[i]=(e384164[i]*e38416k4[i])/e23901[i];
if e23901[i]>0 and e384074[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e384164[i]>=0
    then othpay4[i]=e384164[i]/e23901[i];
if e23901[i]>0 and e384074[i]=4 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2*e23901[i]);
if e23901[i]>0 and e384074[i]=5 and e384164[i]>=0 then othpay4[i]=e384164[i]/(4.3*e23901[i]);
if e23901[i]>0 and e384074[i]=6 and e384164[i]>=0 and e35600[i]>0 then
    othpay4[i]=e384164[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e384074[i]=8 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2.15*e23901[i]);
if e384074[i] in (9, 14) then othpay4[i]=0;
if e384074[i] in (12, 13) and e384164[i]>=0 then do;
    if e38416g4[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225304[i] in (1, 2, 4, 5) and e225004[i]>0 and othpay4[i]>=0 then
            othpay4[i]=othpay4[i]/e225004[i]*e384164[i];
        if e225304[i]=3 and e225004[i]>0 and e38416k4[i]>0 and e225504[i]>0 and othpay4[i]>=0
            then othpay4[i]=othpay4[i]/(e225004[i]*e225504[i])*e384164[i]*e38416k4[i];
        if e225304[i]=3 and e38416k4[i]<=0 then othpay4[i]=-3;
    end;
    else do; /* in all the other cases */
        if e38416i4[i]=1 and e38416ib4[i]>0 then othpay4[i]=(e384164[i]/e38416ib4[i])*60;
        if e38416i4[i]=2 and e38416ib4[i]>0 then othpay4[i]=e384164[i]/e38416ib4[i];
        if e38416i4[i]=3 and e38416ib4[i]>0 and e38416k4[i]>=0 and e23901[i]>0 then
            othpay4[i]=(e384164[i]/e38416ib4[i])*e38416k4[i]/e23901[i];
        if e38416i4[i]=4 and e38416ib4[i]>0 and e23901[i]>0 then othpay4[i]=(e384164[i]/e38416ib4[i])/e23901[i];
        if e38416i4[i]=5 and e38416ib4[i]>0 and e23901[i]>0 then othpay4[i]=(e384164[i]/e38416ib4[i])/(e23901[i]*4.3);
    /* missing value */
        if -4<e38416i4[i]<0 or -4<e38416ib4[i]<=0 then othpay4[i]=-3;
        if e38416i4[i] in (3, 4, 5) and -4<e23901[i]<=0 then othpay4[i]=-3;
        if e38416i4[i]=3 and e38416k4[i]<0 then othpay4[i]=-3;
    end;
end;
if e384074[i] in (1, 2, 3, 4, 5, 6, 7, 8, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and -4<e384164[i]<0 then
    othpay4[i]=e225004[i];
if e384074[i] in (2, 3, 4, 5, 6, 7, 8, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e23901[i]=0 then othpay4[i]=-3;
if e384074[i] in (2, 3, 4, 5, 6, 7, 8, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and -4<e23901[i]<0 then
    othpay4[i]=e23901[i];
if e384074[i]=2 and e38416k4[i] le 0 then othpay4[i]=-3;
if e384074[i]=2 and -4<e38416k4[i]<0 then othpay4[i]=e225504[i];
if e384074[i]=6 and e35600[i] le 0 then othpay4[i]=-3;
if e384074[i]=6 and -4<e35600[i]<0 then othpay4[i]=e35600[i];

/* report non-hourly wage */
if e34428[i]>0 and e384074[i]=1 and e384164[i]>=0 then othpay4[i]=e384164[i];
if e34428[i]>0 and e384074[i]=2 and e38416k4[i]>0 and e384164[i]>=0 then
    othpay4[i]=(e384164[i]*e38416k4[i])/e34428[i];
if e34428[i]>0 and e384074[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e384164[i]>=0
    then othpay4[i]=e384164[i]/e34428[i];
if e34428[i]>0 and e384074[i]=4 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2*e34428[i]);
if e34428[i]>0 and e384074[i]=5 and e384164[i]>=0 then othpay4[i]=e384164[i]/(4.3*e34428[i]);
if e34428[i]>0 and e384074[i]=6 and e384164[i]>=0 and e35600[i]>0 then
    othpay4[i]=(e384164[i]/(e35600[i]*e34428[i]));
if e34428[i]>0 and e384074[i]=8 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2.15*e34428[i]);
if e384074[i] in (12, 13) and e384164[i]>=0 then do;
    if e38416i4[i]=3 and e38416ib4[i]>0 and e38416k4[i]>=0 and e34428[i]>0 then
        othpay4[i]=(e384164[i]/e38416ib4[i])*e38416k4[i]/e34428[i];

```

Appendix 2: Employment Variable Creation

```

if e38416i4[i]=4 and e38416ib4[i]>0 and e34428[i]>0 then othpay4[i]=(e384164[i]/e38416ib4[i])/e34428[i];
if e38416i4[i]=5 and e38416ib4[i]>0 and e34428[i]>0 then othpay4[i]=(e384164[i]/e38416ib4[i])/(e34428[i]*4.3);
/* missing value */
if e38416i4[i] in (3,4,5) and -4<e34428[i]<=0 then othpay4[i]=-3;
end;

if e384074[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34428[i]=0 then othpay4[i]=-3;
if e384074[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34428[i]<0 then
    othpay4[i]=e34428[i];

end;

/*without overtime at the beginning*/
if e3800b[i]=1 then do;

    /*report hourly wage at the beginning*/
    if e23901[i]>0 and e384074[i]=1 and e384164[i]>=0 then othpay4[i]=e384164[i];
    if e23901[i]>0 and e384074[i]=2 and e38416k4[i]>0 and e384164[i]>=0 then
        othpay4[i]=(e384164[i]*e38416k4[i])/e23901[i];
    if e23901[i]>0 and e384074[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384164[i]>=0
        then othpay4[i]=e384164[i]/e23901[i];
    if e23901[i]>0 and e384074[i]=4 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2*e23901[i]);
    if e23901[i]>0 and e384074[i]=5 and e384164[i]>=0 then othpay4[i]=e384164[i]/(4.3*e23901[i]);
    if e23901[i]>0 and e384074[i]=6 and e384164[i]>=0 and e35600[i]>0 then
        othpay4[i]=e384164[i]/(e35600[i]*e23901[i]);
    if e23901[i]>0 and e384074[i]=8 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2.15*e23901[i]);
    if e384074[i] in (9,14) then othpay4[i]=0;
    if e384074[i] in (12,13) and e384164[i]>=0 then do;
        if e38416g4[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
            if e225304[i] in (1,2,4,5) and e225004[i]>0 and othpay4[i]>=0 then
                othpay4[i]=othpay4[i]/e225004[i]*e384164[i];
            if e225304[i]=3 and e225004[i]>0 and e38416k4[i]>0 and e225504[i]>0 and othpay4[i]>=0
                then othpay4[i]=othpay4[i]/(e225004[i]*e225504[i])*e384164[i]*e38416k4[i];
            if e225304[i]=3 and e38416k4[i]<=0 then othpay4[i]=-3;
        end;
        else do; /*in all the other cases */
            if e38416i4[i]=1 and e38416ib4[i]>0 then othpay4[i]=(e384164[i]/e38416ib4[i])*60;
            if e38416i4[i]=2 and e38416ib4[i]>0 then othpay4[i]=e384164[i]/e38416ib4[i];
            if e38416i4[i]=3 and e38416ib4[i]>0 and e38416k4[i]>=0 and e23901[i]>0 then
                othpay4[i]=(e384164[i]/e38416ib4[i])*e38416k4[i]/e23901[i];
            if e38416i4[i]=4 and e38416ib4[i]>0 and e23901[i]>0 then othpay4[i]=(e384164[i]/e38416ib4[i])/e23901[i];
            if e38416i4[i]=5 and e38416ib4[i]>0 and e23901[i]>0 then othpay4[i]=(e384164[i]/e38416ib4[i])/(e23901[i]*4.3);
            /* missing value */
            if -4<e38416i4[i]<0 or -4<e38416ib4[i]<=0 then othpay4[i]=-3;
            if e38416i4[i] in (3,4,5) and -4<e23901[i]<=0 then othpay4[i]=-3;
            if e38416i4[i]=3 and e38416k4[i]<0 then othpay4[i]=-3;
        end;
    end;
    if e384074[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e384164[i]<0 then
        othpay4[i]=e225004[i];
    if e384074[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay4[i]=-3;
    if e384074[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
        othpay4[i]=e23901[i];
    if e384074[i]=2 and e38416k4[i]<0 then othpay4[i]=-3;
    if e384074[i]=2 and -4<e38416k4[i]<0 then othpay4[i]=e225504[i];
    if e384074[i]=6 and e35600[i]<0 then othpay4[i]=-3;

```

Appendix 2: Employment Variable Creation

```

if e384074[i]=6 and -4<e35600[i]<0 then othpay4[i]=e35600[i];

/*report non-hourly wage*/
if e34402[i]>0 and e384074[i]=1 and e384164[i]>=0 then othpay4[i]=e384164[i];
if e34402[i]>0 and e384074[i]=2 and e38416k4[i]>0 and e384164[i]>=0 then
    othpay4[i]=(e384164[i]*e38416k4[i])/e34402[i];
if e34402[i]>0 and e384074[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e384164[i]>=0
    then othpay4[i]=e384164[i]/e34402[i];
if e34402[i]>0 and e384074[i]=4 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2*e34402[i]);
if e34402[i]>0 and e384074[i]=5 and e384164[i]>=0 then othpay4[i]=e384164[i]/(4.3*e34402[i]);
if e34402[i]>0 and e384074[i]=6 and e384164[i]>=0 and e35600[i]>0 then
    othpay4[i]=e384164[i]/(e35600[i]*e34402[i]);
if e34402[i]>0 and e384074[i]=8 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2.15*e34402[i]);
if e34402[i]>0 and e384074[i] in (9, 7, 12, 13, 114, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28) then othpf4[i]=othpf4[i]+1;
if e384074[i] in (12, 13) and e384164[i]>=0 then do;
    if e38416i4[i]=3 and e38416ib4[i]>0 and e38416k4[i]>0 and e34402[i]>0 then
        othpay4[i]=(e384164[i]/e38416ib4[i])*e38416k4[i]/e34402[i];
    if e38416i4[i]=4 and e38416ib4[i]>0 and e34402[i]>0 then othpay4[i]=(e384164[i]/e38416ib4[i])/e34402[i];
    if e38416i4[i]=5 and e38416ib4[i]>0 and e34402[i]>0 then othpay4[i]=(e384164[i]/e38416ib4[i])/(e34402[i]*4.3);
/* missing value */
    if e38416i4[i] in (3, 4, 5) and -4<e34402[i]<=0 then othpay4[i]=-3;
end;

if e384074[i] in (2, 3, 4, 5, 6, 7, 8, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e34402[i]=0 then othpay4[i]=-3;
if e384074[i] in (2, 3, 4, 5, 6, 7, 8, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and -4<e34402[i]<0 then
    othpay4[i]=e34402[i];
end;

/*for others*/
/*with overtime at the beginning*/
if e38102[i]>0 and e38102[i] ne 1 and e38102[i] ne 3 then do;

/* report hourly wage at the beginning*/
if e23901[i]>0 and e384075[i]=1 and e384165[i]>=0 then othpay5[i]=e384165[i];
if e23901[i]>0 and e384075[i]=2 and e38416k5[i]>0 and e384165[i]>=0 then
    othpay5[i]=(e384165[i]*e38416k5[i])/e23901[i];
if e23901[i]>0 and e384075[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e384165[i]>=0
    then othpay5[i]=e384165[i]/e23901[i];
if e23901[i]>0 and e384075[i]=4 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2*e23901[i]);
if e23901[i]>0 and e384075[i]=5 and e384165[i]>=0 then othpay5[i]=e384165[i]/(4.3*e23901[i]);
if e23901[i]>0 and e384075[i]=6 and e384165[i]>=0 and e35600[i]>0 then
    othpay5[i]=e384165[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e384075[i]=8 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2.15*e23901[i]);
if e384075[i] in (9, 14) then othpay5[i]=0;
if e384075[i] in (12, 13) and e384165[i]>=0 then do;
    if e38416g5[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225305[i] in (1, 2, 4, 5) and e225005[i]>0 and othpay5[i]>=0 then
            othpay5[i]=othpay5[i]/e225005[i]*e384165[i];
        if e225305[i]=3 and e225005[i]>0 and e38416k5[i]>0 and e225505[i]>0 and othpay5[i]>=0
            then othpay5[i]=othpay5[i]/(e225005[i]*e225505[i])*e384165[i]*e38416k5[i];
        if e225305[i]=3 and e38416k5[i]<=0 then othpay5[i]=-3;
    end;
else do; /*in all the other cases */
    if e38416i5[i]=1 and e38416ib5[i]>0 then othpay5[i]=(e384165[i]/e38416ib5[i])*60;
    if e38416i5[i]=2 and e38416ib5[i]>0 then othpay5[i]=e384165[i]/e38416ib5[i];

```

```

if e38416i5[i]=3 and e38416ib5[i]>0 and e38416k5[i]>=0 and e23901[i]>0 then
    othpay5[i]=(e384165[i]/e38416ib5[i])*e38416k5[i]/e23901[i];
if e38416i5[i]=4 and e38416ib5[i]>0 and e23901[i]>0 then othpay5[i]=(e384165[i]/e38416ib5[i])/e23901[i];
if e38416i5[i]=5 and e38416ib5[i]>0 and e23901[i]>0 then othpay5[i]=(e384165[i]/e38416ib5[i])/(e23901[i]*4.3);
/* missing value */
if -4<e38416i5[i]<0 or -4<e38416ib5[i]<=0 then othpay5[i]=-3;
if e38416i5[i] in (3,4,5) and -4<e23901[i]<=0 then othpay5[i]=-3;
if e38416i5[i]=3 and e38416k5[i]<0 then othpay5[i]=-3;
end;
end;

if e384075[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e384165[i]<0 then
    othpay5[i]=e225005[i];
if e384075[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay5[i]=-3;
if e384075[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay5[i]=e23901[i];
if e384075[i]=2 and e38416k5[i] le 0 then othpay5[i]=-3;
if e384075[i]=2 and -4<e38416k5[i]<0 then othpay5[i]=e225505[i];
if e384075[i]=6 and e35600[i] le 0 then othpay5[i]=-3;
if e384075[i]=6 and -4<e35600[i]<0 then othpay5[i]=e35600[i];

/*report non-hourly wage*/
if e34428[i]>0 and e384075[i]=1 and e384165[i]>=0 then othpay5[i]=e384165[i];
if e34428[i]>0 and e384075[i]=2 and e38416k5[i]>0 and e384165[i]>=0 then
    othpay5[i]=(e384165[i]*e38416k5[i])/e34428[i];
if e34428[i]>0 and e384075[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384165[i]>=0
    then othpay5[i]=e384165[i]/e34428[i];
if e34428[i]>0 and e384075[i]=4 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2*e34428[i]);
if e34428[i]>0 and e384075[i]=5 and e384165[i]>=0 then othpay5[i]=e384165[i]/(4.3*e34428[i]);
if e34428[i]>0 and e384075[i]=6 and e384165[i]>=0 and e35600[i]>0 then
    othpay5[i]=e384165[i]/(e35600[i]*e34428[i]);
if e34428[i]>0 and e384075[i]=8 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2.15*e34428[i]);
if e384075[i] in (12,13) and e384165[i]>=0 then do;
    if e38416i5[i]=3 and e38416ib5[i]>0 and e38416k5[i]>=0 and e34428[i]>0 then
        othpay5[i]=(e384165[i]/e38416ib5[i])*e38416k5[i]/e34428[i];
        if e38416i5[i]=4 and e38416ib5[i]>0 and e34428[i]>0 then othpay5[i]=(e384165[i]/e38416ib5[i])/e34428[i];
        if e38416i5[i]=5 and e38416ib5[i]>0 and e34428[i]>0 then othpay5[i]=(e384165[i]/e38416ib5[i])/(e34428[i]*4.3);
    /* missing value */
    if e38416i5[i] in (3,4,5) and -4<e34428[i]<=0 then othpay5[i]=-3;
end;

if e384075[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34428[i]=0 then othpay5[i]=-3;
if e384075[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34428[i]<0 then
    othpay5[i]=e34428[i];

end;

/*without overtime at the beginning*/
if e3800b[i]=1 then do;

/* report hourly wage at the beginning*/
if e23901[i]>0 and e384075[i]=1 and e384165[i]>=0 then othpay5[i]=e384165[i];
if e23901[i]>0 and e384075[i]=2 and e38416k5[i]>0 and e384165[i]>=0 then
    othpay5[i]=(e384165[i]*e38416k5[i])/e23901[i];
if e23901[i]>0 and e384075[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384165[i]>=0
    then othpay5[i]=e384165[i]/e23901[i];
if e23901[i]>0 and e384075[i]=4 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2*e23901[i]);

```

Appendix 2: Employment Variable Creation

```

if e23901[i]>0 and e384075[i]=5 and e384165[i]>=0 then othpay5[i]=e384165[i]/(4.3*e23901[i]);
if e23901[i]>0 and e384075[i]=6 and e384165[i]>=0 and e35600[i]>0 then
    othpay5[i]=e384165[i]/(e35600[i]*e23901[i]);
if e23901[i]>0 and e384075[i]=8 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2.15*e23901[i]);
if e384075[i] in (9,14) then othpay5[i]=0;
if e384075[i] in (12,13) and e384165[i]>=0 then do;
    if e38416g5[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225305[i] in (1,2,4,5) and e225005[i]>0 and othpay5[i]>=0 then
            othpay5[i]=othpay5[i]/e225005[i]*e384165[i];
        if e225305[i]=3 and e225005[i]>0 and e38416k5[i]>0 and e225505[i]>0 and othpay5[i]>=0
            then othpay5[i]=othpay5[i]/(e225005[i]*e225505[i])*e384165[i]*e38416k5[i];
        if e225305[i]=3 and e38416k5[i]<=0 then othpay5[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38416i5[i]=1 and e38416ib5[i]>0 then othpay5[i]=(e384165[i]/e38416ib5[i])*60;
        if e38416i5[i]=2 and e38416ib5[i]>0 then othpay5[i]=e384165[i]/e38416ib5[i];
        if e38416i5[i]=3 and e38416ib5[i]>0 and e38416k5[i]>=0 and e23901[i]>0 then
            othpay5[i]=(e384165[i]/e38416ib5[i])*e38416k5[i]/e23901[i];
        if e38416i5[i]=4 and e38416ib5[i]>0 and e23901[i]>0 then othpay5[i]=(e384165[i]/e38416ib5[i])/e23901[i];
        if e38416i5[i]=5 and e38416ib5[i]>0 and e23901[i]>0 then othpay5[i]=(e384165[i]/e38416ib5[i])/(e23901[i]*4.3);
    /* missing value */
        if -4<e38416i5[i]<0 or -4<e38416ib5[i]<=0 then othpay5[i]=-3;
        if e38416i5[i] in (3,4,5) and -4<e23901[i]<=0 then othpay5[i]=-3;
        if e38416i5[i]=3 and e38416k5[i]<0 then othpay5[i]=-3;
    end;
end;

if e384075[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e384165[i]<0 then
    othpay5[i]=e225005[i];
if e384075[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e23901[i]=0 then othpay5[i]=-3;
if e384075[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e23901[i]<0 then
    othpay5[i]=e23901[i];
if e384075[i]=2 and e38416k5[i] le 0 then othpay5[i]=-3;
if e384075[i]=2 and -4<e38416k5[i]<0 then othpay5[i]=e225505[i];
if e384075[i]=6 and e35600[i] le 0 then othpay5[i]=-3;
if e384075[i]=6 and -4<e35600[i]<0 then othpay5[i]=e35600[i];

/*report non-hourly wage*/
if e34402[i]>0 and e384075[i]=1 and e384165[i]>=0 then othpay5[i]=e384165[i];
if e34402[i]>0 and e384075[i]=2 and e38416k5[i]>0 and e384165[i]>=0 then
    othpay5[i]=(e384165[i]*e38416k5[i])/e34402[i];
if e34402[i]>0 and e384075[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384165[i]>=0
    then othpay5[i]=e384165[i]/e34402[i];
if e34402[i]>0 and e384075[i]=4 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2*e34402[i]);
if e34402[i]>0 and e384075[i]=5 and e384165[i]>=0 then othpay5[i]=e384165[i]/(4.3*e34402[i]);
if e34402[i]>0 and e384075[i]=6 and e384165[i]>=0 and e35600[i]>0 then
    othpay5[i]=e384165[i]/(e35600[i]*e34402[i]);
if e34402[i]>0 and e384075[i]=8 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2.15*e34402[i]);
if e34402[i]>0 and e384075[i] in (9,7,12,13,14,15,16,17,18,19,21,22,23,24,25,26,28) then othpf5[i]=othpf5[i]+1;
if e384075[i] in (12,13) and e384165[i]>=0 then do;
    if e38416i5[i]=3 and e38416ib5[i]>0 and e38416k5[i]>=0 and e34402[i]>0 then
        othpay5[i]=(e384165[i]/e38416ib5[i])*e38416k5[i]/e34402[i];
    if e38416i5[i]=4 and e38416ib5[i]>0 and e34402[i]>0 then othpay5[i]=(e384165[i]/e38416ib5[i])/e34402[i];
    if e38416i5[i]=5 and e38416ib5[i]>0 and e34402[i]>0 then othpay5[i]=(e384165[i]/e38416ib5[i])/(e34402[i]*4.3);
/* missing value */
    if e38416i5[i] in (3,4,5) and -4<e34402[i]<=0 then othpay5[i]=-3;
end;

```

```

if e384075[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e34402[i]=0 then othpay5[i]=-3;
if e384075[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e34402[i]<0 then
    othpay5[i]=e34402[i];

end;
end;

/** case vi. more than one compensation at the beginning, diff no. of hours. ***/

if e38330[i]=1 and e20700[i]=1 and (e38102[i]=1 or e38102[i]=3) then do;

/* for tips*/

if e38103[i]>0 and e384071[i]=1 and e384161[i]>=0 then othpay1[i]=e384161[i];
if e38103[i]>0 and e384071[i]=2 and e38416k1[i]>0 and e384161[i]>=0 then
    othpay1[i]=(e384161[i]*e38416k1[i])/e38103[i];
if e38103[i]>0 and e384071[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384161[i]>=0
    then othpay1[i]=e384161[i]/e38103[i];
if e38103[i]>0 and e384071[i]=4 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2*e38103[i]);
if e38103[i]>0 and e384071[i]=5 and e384161[i]>=0 then othpay1[i]=e384161[i]/(4.3*e38103[i]);
if e38103[i]>0 and e384071[i]=6 and e384161[i]>=0 and e35600[i]>0 then
    othpay1[i]=e384161[i]/(e35600[i]*e38103[i]);
if e38103[i]>0 and e384071[i]=8 and e384161[i]>=0 then othpay1[i]=e384161[i]/(2.15*e38103[i]);
if e384071[i] in (9, 14) then othpay1[i]=0;
if e384071[i] in (12,13) and e384161[i]>=0 then do;
    if e38416g1[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225301[i] in (1,2,4,5) and e225001[i]>0 and othpay1[i]>=0 then
            othpay1[i]=othpay1[i]/e225001[i]*e384161[i];
        if e225301[i]=3 and e225001[i]>0 and e38416k1[i]>0 and e225501[i]>0 and othpay1[i]>=0
            then othpay1[i]=othpay1[i]/(e225001[i]*e225501[i])*e384161[i]*e38416k1[i];
        if e225301[i]=3 and e38416k1[i]<=0 then othpay1[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38416i1[i]=1 and e38416ib1[i]>0 then othpay1[i]=(e384161[i]/e38416ib1[i])*60;
        if e38416i1[i]=2 and e38416ib1[i]>0 then othpay1[i]=e384161[i]/e38416ib1[i];
        if e38416i1[i]=3 and e38416ib1[i]>0 and e38416k1[i]>=0 and e38103[i]>0 then
            othpay1[i]=(e384161[i]/e38416ib1[i])*e38416k1[i]/e38103[i];
        if e38416i1[i]=4 and e38416ib1[i]>0 and e38103[i]>0 then othpay1[i]=(e384161[i]/e38416ib1[i])/e38103[i];
        if e38416i1[i]=5 and e38416ib1[i]>0 and e38103[i]>0 then othpay1[i]=(e384161[i]/e38416ib1[i])/(e38103[i]*4.3);
    /* missing value */
        if -4<e38416i1[i]<0 or -4<e38416ib1[i]<=0 then othpay1[i]=-3;
        if e38416i1[i] in (3,4,5) and -4<e38103[i]<=0 then othpay1[i]=-3;
        if e38416i1[i]=3 and e38416k1[i]<0 then othpay1[i]=-3;
    end;
end;
if e384071[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e384161[i]<0 then
    othpay1[i]=e225001[i];
if e384071[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e38103[i]=0 then othpay1[i]=-3;
if e384071[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38103[i]<0 then
    othpay1[i]=e38103[i];
if e384071[i]=2 and e38416k1[i] le 0 then othpay1[i]=-3;
if e384071[i]=2 and -4<e38416k1[i]<0 then othpay1[i]=e225501[i];
if e384071[i]=6 and e35600[i] le 0 then othpay1[i]=-3;
if e384071[i]=6 and -4<e35600[i]<0 then othpay1[i]=e35600[i];

```

Appendix 2: Employment Variable Creation

/*for commissions*/

```

if e38103[i]>0 and e384072[i]=1 and e384162[i]>=0 then othpay2[i]=e384162[i];
if e38103[i]>0 and e384072[i]=2 and e38416k2[i]>0 and e384162[i]>=0 then
    othpay2[i]=(e384162[i]*e38416k2[i])/e38103[i];
if e38103[i]>0 and e384072[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e384162[i]>=0
    then othpay2[i]=e384162[i]/e38103[i];
if e38103[i]>0 and e384072[i]=4 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2*e38103[i]);
if e38103[i]>0 and e384072[i]=5 and e384162[i]>=0 then othpay2[i]=e384162[i]/(4.3*e38103[i]);
if e38103[i]>0 and e384072[i]=6 and e384162[i]>=0 and e35600[i]>0 then
    othpay2[i]=e384162[i]/(e35600[i]*e38103[i]);
if e38103[i]>0 and e384072[i]=8 and e384162[i]>=0 then othpay2[i]=e384162[i]/(2.15*e38103[i]);
if e384071[i] in (9, 14) then othpay1[i]=0;
if e384072[i] in (12, 13) and e384162[i]>=0 then do;
    if e38416g2[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225302[i] in (1, 2, 4, 5) and e225002[i]>0 and othpay2[i]>=0 then
            othpay2[i]=othpay2[i]/e225002[i]*e384162[i];
        if e225302[i]=3 and e225002[i]>0 and e38416k2[i]>0 and e225502[i]>0 and othpay2[i]>=0
            then othpay2[i]=othpay2[i]/(e225002[i]*e225502[i])*e384162[i]*e38416k2[i];
        if e225302[i]=3 and e38416k2[i]<=0 then othpay2[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38416i2[i]=1 and e38416ib2[i]>0 then othpay2[i]=(e384162[i]/e38416ib2[i])*60;
        if e38416i2[i]=2 and e38416ib2[i]>0 then othpay2[i]=e384162[i]/e38416ib2[i];
        if e38416i2[i]=3 and e38416ib2[i]>0 and e38416k2[i]>=0 and e38103[i]>0 then
            othpay2[i]=(e384162[i]/e38416ib2[i])*e38416k2[i]/e38103[i];
        if e38416i2[i]=4 and e38416ib2[i]>0 and e38103[i]>0 then othpay2[i]=(e384162[i]/e38416ib2[i])/e38103[i];
        if e38416i2[i]=5 and e38416ib2[i]>0 and e38103[i]>0 then othpay2[i]=(e384162[i]/e38416ib2[i])/(e38103[i]*4.3);
    /* missing value */
        if -4<e38416i2[i]<0 or -4<e38416ib2[i]<=0 then othpay2[i]=-3;
        if e38416i2[i] in (3, 4, 5) and -4<e38103[i]<=0 then othpay2[i]=-3;
        if e38416i2[i]=3 and e38416k2[i]<0 then othpay2[i]=-3;
    end;
end;
if e384072[i] in (1, 2, 3, 4, 5, 6, 7, 8, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and -4<e384162[i]<0 then
    othpay2[i]=e225002[i];
if e384072[i] in (2, 3, 4, 5, 6, 7, 8, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e38103[i]=0 then othpay2[i]=-3;
if e384072[i] in (2, 3, 4, 5, 6, 7, 8, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and -4<e38103[i]<0 then
    othpay2[i]=e38103[i];
if e384072[i]=2 and e38416k2[i] le 0 then othpay2[i]=-3;
if e384072[i]=2 and -4<e38416k2[i]<0 then othpay2[i]=e225502[i];
if e384072[i]=6 and e35600[i] le 0 then othpay2[i]=-3;
if e384072[i]=6 and -4<e35600[i]<0 then othpay2[i]=e35600[i];

```

/*for bonuses*/

```

if e38103[i]>0 and e384073[i]=1 and e384163[i]>=0 then othpay3[i]=e384163[i];
if e38103[i]>0 and e384073[i]=2 and e38416k3[i]>0 and e384163[i]>=0 then
    othpay3[i]=(e384163[i]*e38416k3[i])/e38103[i];
if e38103[i]>0 and e384073[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 999, -1, -2) and e384163[i]>=0
    then othpay3[i]=e384163[i]/e38103[i];
if e38103[i]>0 and e384073[i]=4 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2*e38103[i]);
if e38103[i]>0 and e384073[i]=5 and e384163[i]>=0 then othpay3[i]=e384163[i]/(4.3*e38103[i]);
if e38103[i]>0 and e384073[i]=6 and e384163[i]>=0 and e35600[i]>0 then
    othpay3[i]=e384163[i]/(e35600[i]*e38103[i]);
if e38103[i]>0 and e384073[i]=8 and e384163[i]>=0 then othpay3[i]=e384163[i]/(2.15*e38103[i]);

```

```

if e384073[i] in (9, 14) then othpay3[i]=0;
if e384073[i] in (12,13) and e384163[i]>=0 then do;
  if e38416g3[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
    if e225303[i] in (1,2,4,5) and e225003[i]>0 and othpay3[i]>=0 then
      othpay3[i]=othpay3[i]/e225003[i]*e384163[i];
    if e225303[i]=3 and e225003[i]>0 and e38416k3[i]>0 and e225503[i]>0 and othpay3[i]>=0
      then othpay3[i]=othpay3[i]/(e225003[i]*e225503[i])*e384163[i]*e38416k3[i];
    if e225303[i]=3 and e38416k3[i]<=0 then othpay3[i]=-3;
  end;
  else do; /*in all the other cases */
    if e38416i3[i]=1 and e38416ib3[i]>0 then othpay3[i]=(e384163[i]/e38416ib3[i])*60;
    if e38416i3[i]=2 and e38416ib3[i]>0 then othpay3[i]=e384163[i]/e38416ib3[i];
    if e38416i3[i]=3 and e38416ib3[i]>0 and e38416k3[i]>=0 and e38103[i]>0 then
      othpay3[i]=(e384163[i]/e38416ib3[i])*e38416k3[i]/e38103[i];
    if e38416i3[i]=4 and e38416ib3[i]>0 and e38103[i]>0 then othpay3[i]=(e384163[i]/e38416ib3[i])/e38103[i];
    if e38416i3[i]=5 and e38416ib3[i]>0 and e38103[i]>0 then othpay3[i]=(e384163[i]/e38416ib3[i])/(e38103[i]*4.3);
    /* missing value */
    if -4<e38416i3[i]<0 or -4<e38416ib3[i]<=0 then othpay3[i]=-3;
    if e38416i3[i] in (3,4,5) and -4<e38103[i]<=0 then othpay3[i]=-3;
    if e38416i3[i]=3 and e38416k3[i]<0 then othpay3[i]=-3;
  end;
end;

if e384073[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e384163[i]<0 then
  othpay3[i]=e225003[i];
if e384073[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e38103[i]=0 then othpay3[i]=-3;
if e384073[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38103[i]<0 then
  othpay3[i]=e38103[i];
if e384073[i]=2 and e38416k3[i] le 0 then othpay3[i]=-3;
if e384073[i]=2 and -4<e38416k3[i]<0 then othpay3[i]=e225503[i];
if e384073[i]=6 and e35600[i] le 0 then othpay3[i]=-3;
if e384073[i]=6 and -4<e35600[i]<0 then othpay3[i]=e35600[i];

/*for incentive pay*/

if e38103[i]>0 and e384074[i]=1 and e384164[i]>=0 then othpay4[i]=e384164[i];
if e38103[i]>0 and e384074[i]=2 and e38416k4[i]>0 and e384164[i]>=0 then
  othpay4[i]=(e384164[i]*e38416k4[i])/e38103[i];
if e38103[i]>0 and e384074[i] in (3, 7,15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384164[i]>=0
  then othpay4[i]=e384164[i]/e38103[i];
if e38103[i]>0 and e384074[i]=4 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2*e38103[i]);
if e38103[i]>0 and e384074[i]=5 and e384164[i]>=0 then othpay4[i]=e384164[i]/(4.3*e38103[i]);
if e38103[i]>0 and e384074[i]=6 and e384164[i]>=0 and e35600[i]>0 then
  othpay4[i]=e384164[i]/(e35600[i]*e38103[i]);
if e38103[i]>0 and e384074[i]=8 and e384164[i]>=0 then othpay4[i]=e384164[i]/(2.15*e38103[i]);
if e384074[i] in (9, 14) then othpay4[i]=0;
if e384074[i] in (12,13) and e384164[i]>=0 then do;
  if e38416g4[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
    if e225304[i] in (1,2,4,5) and e225004[i]>0 and othpay4[i]>=0 then
      othpay4[i]=othpay4[i]/e225004[i]*e384164[i];
    if e225304[i]=3 and e225004[i]>0 and e38416k4[i]>0 and e225504[i]>0 and othpay4[i]>=0
      then othpay4[i]=othpay4[i]/(e225004[i]*e225504[i])*e384164[i]*e38416k4[i];
    if e225304[i]=3 and e38416k4[i]<=0 then othpay4[i]=-3;
  end;
  else do; /*in all the other cases */
    if e38416i4[i]=1 and e38416ib4[i]>0 then othpay4[i]=(e384164[i]/e38416ib4[i])*60;
    if e38416i4[i]=2 and e38416ib4[i]>0 then othpay4[i]=e384164[i]/e38416ib4[i];

```

Appendix 2: Employment Variable Creation

```

if e38416i4[i]=3 and e38416ib4[i]>0 and e38416k4[i]>=0 and e38103[i]>0 then
    othpay4[i]=(e384164[i]/e38416ib4[i])*e38416k4[i]/e38103[i];
if e38416i4[i]=4 and e38416ib4[i]>0 and e38103[i]>0 then othpay4[i]=(e384164[i]/e38416ib4[i])/e38103[i];
if e38416i4[i]=5 and e38416ib4[i]>0 and e38103[i]>0 then othpay4[i]=(e384164[i]/e38416ib4[i])/(e38103[i]*4.3);
/* missing value */
if -4<e38416i4[i]<0 or -4<e38416ib4[i]<=0 then othpay4[i]=-3;
if e38416i4[i] in (3,4,5) and -4<e38103[i]<=0 then othpay4[i]=-3;
if e38416i4[i]=3 and e38416k4[i]<0 then othpay4[i]=-3;
end;
end;

if e384074[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e384164[i]<0 then
    othpay4[i]=e225004[i];
if e384074[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e38103[i]=0 then othpay4[i]=-3;
if e384074[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38103[i]<0 then
    othpay4[i]=e38103[i];
if e384074[i]=2 and e38416k4[i] le 0 then othpay4[i]=-3;
if e384074[i]=2 and -4<e38416k4[i]<0 then othpay4[i]=e225504[i];
if e384074[i]=6 and e35600[i] le 0 then othpay4[i]=-3;
if e384074[i]=6 and -4<e35600[i]<0 then othpay4[i]=e35600[i];

/*for others*/
if e38103[i]>0 and e384075[i]=1 and e384165[i]>=0 then othpay5[i]=e384165[i];
if e38103[i]>0 and e384075[i]=2 and e38416k5[i]>0 and e384165[i]>=0 then
    othpay5[i]=(e384165[i]*e38416k5[i])/e38103[i];
if e38103[i]>0 and e384075[i] in (3, 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28,999,-1,-2) and e384165[i]>=0
    then othpay5[i]=e384165[i]/e38103[i];
if e38103[i]>0 and e384075[i]=4 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2*e38103[i]);
if e38103[i]>0 and e384075[i]=5 and e384165[i]>=0 then othpay5[i]=e384165[i]/(4.3*e38103[i]);
if e38103[i]>0 and e384075[i]=6 and e384165[i]>=0 and e35600[i]>0 then
    othpay5[i]=e384165[i]/(e35600[i]*e38103[i]);
if e38103[i]>0 and e384075[i]=8 and e384165[i]>=0 then othpay5[i]=e384165[i]/(2.15*e38103[i]);
if e384075[i] in (9, 14) then othpay5[i]=0;
if e384075[i] in (12,13) and e384165[i]>=0 then do;
    if e38416g5[i]=1 then do; /* if the speed info. given at the beginning doesn't change. */
        if e225305[i] in (1,2,4,5) and e225005[i]>0 and othpay5[i]>=0 then
            othpay5[i]=othpay5[i]/e225005[i]*e384165[i];
        if e225305[i]=3 and e225005[i]>0 and e38416k5[i]>0 and e225505[i]>0 and othpay5[i]>=0
            then othpay5[i]=othpay5[i]/(e225005[i]*e225505[i])*e384165[i]*e38416k5[i];
        if e225305[i]=3 and e38416k5[i]<=0 then othpay5[i]=-3;
    end;
    else do; /*in all the other cases */
        if e38416i5[i]=1 and e38416ib5[i]>0 then othpay5[i]=(e384165[i]/e38416ib5[i])*60;
        if e38416i5[i]=2 and e38416ib5[i]>0 then othpay5[i]=e384165[i]/e38416ib5[i];
        if e38416i5[i]=3 and e38416ib5[i]>0 and e38416k5[i]>=0 and e38103[i]>0 then
            othpay5[i]=(e384165[i]/e38416ib5[i])*e38416k5[i]/e38103[i];
        if e38416i5[i]=4 and e38416ib5[i]>0 and e38103[i]>0 then othpay5[i]=(e384165[i]/e38416ib5[i])/e38103[i];
        if e38416i5[i]=5 and e38416ib5[i]>0 and e38103[i]>0 then othpay5[i]=(e384165[i]/e38416ib5[i])/(e38103[i]*4.3);
    /* missing value */
        if -4<e38416i5[i]<0 or -4<e38416ib5[i]<=0 then othpay5[i]=-3;
        if e38416i5[i] in (3,4,5) and -4<e38103[i]<=0 then othpay5[i]=-3;
        if e38416i5[i]=3 and e38416k5[i]<0 then othpay5[i]=-3;
    end;
end;

```

Appendix 2: Employment Variable Creation

```

if e384075[i] in (1,2,3,4,5,6,7,8,12,13,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e384165[i]<0 then
    othpay5[i]=e225005[i];
if e384075[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and e38103[i]=0 then othpay5[i]=-3;
if e384075[i] in (2,3,4,5,6,7,8,15,16,17,18,19,21,22,23,24,25,26,28,999,-1,-2) and -4<e38103[i]<0 then
    othpay5[i]=e38103[i];
if e384075[i]=2 and e38416k5[i] le 0 then othpay5[i]=-3;
if e384075[i]=2 and -4<e38416k5[i]<0 then othpay5[i]=e225505[i];
if e384075[i]=6 and e35600[i] le 0 then othpay5[i]=-3;
if e384075[i]=6 and -4<e35600[i]<0 then othpay5[i]=e35600[i];
end;
end;
end;

***** overall end hourly compensation *****

do i=1 to 9;
if e37901b[i]=1 or e59900[i]=1 then do;
    hrcomp[i]=0;
    if hrwg[i] ge 0 then hrcomp[i]=hrcomp[i]+hrwg[i];
    if ot[i] ge 0 then hrcomp[i]=hrcomp[i]+ot[i];
    if othpay1[i] ge 0 then hrcomp[i]=hrcomp[i]+othpay1[i];
    if othpay2[i] ge 0 then hrcomp[i]=hrcomp[i]+othpay2[i];
    if othpay3[i] ge 0 then hrcomp[i]=hrcomp[i]+othpay3[i];
    if othpay4[i] ge 0 then hrcomp[i]=hrcomp[i]+othpay4[i];
    if othpay5[i] ge 0 then hrcomp[i]=hrcomp[i]+othpay5[i];
    if -4<hrwg[i]<0 or -4<ot[i]<0 or -4<othpay1[i]<0 or -4<othpay2[i]<0 or -4<othpay3[i]<0 or -4<othpay4[i]<0 or -
        4<othpay5[i]<0 then hrcomp[i]=-3;
    if hrwg[i]=-4 then hrcomp[i]=-4;
end;
end;

/* fix the hourly compensation because the respondents may understand the overtime question differently. 1) if overtime
   pay is less than or equal to the hourly wage, we assume the resp. consider the overtime as something above the
   hourly wage. so the hourly comp. should include both hourly wage and overtime pay. 2) if overtime pay is
   greater than the hourly wage, we assume the respondents consider the overtime pay as a pay including hourly
   wage. so the hourly comp. only include the overtime pay without the hourly wage. */

do i=1 to 9;
    if hrwg[i] gt 0 and ot[i] gt hrwg[i] and hrcomp[i] gt 0 then hrcomp[i]=hrcomp[i]-hrwg[i];
end;

***          fix round 4 hourly compensation calculation. I will call the hourly wage, hourly overtime and hourly
other compensation calculated before as previous computed hourly wage,etc. To fix it, I will first find the
overtime hours since I only need to fix those people who have overtime hours. For the people who have
overtime hours, the fixed total hourly comp. equals:
    ( (hrly wage)*(reg. hours) + (prev. hrly ot)*(ot hrs) + (prev. other hrly comp)*(reg. hrs) ) / total hrs.
All the "hours" in the above equation refers to "weekly hours".
There are 2 exceptions to the above equation. When the time unit to other compensation is hrly or per item (and minute
or hour as unit for each item), we multiply (prev. other hrly comp) by (total hrs) instead of (reg hrs). This term
should give us the weekly other compensation. since previously we use regular hours to calculate hrly other
compensation regardless, we use reg hrs in the equation above to recover the previous calculated hrly other
compensation to the weekly other compensation. but if the reported time unit is hrly, we don't have to recover
anything and just use the total weekly hours. */

array regrhr regrhr1-regrhr9;
array ttlhr ttlhr1-ttlhr9;
array pay2hr pay2hr1-pay2hr9;
array othrrhr othrrhr1-othrrhr9;
array pay1hr pay1hr1-pay1hr9;
array pay3hr pay3hr1-pay3hr9;

```

```

array pay4hr pay4hr1-pay4hr9;           array pay5hr pay5hr1-pay5hr9;
array newcomp newcomp1-newcomp9;         new=0;

array hrwgtp hrwgtp1-hrwgtp9;          array ottp ottp1-ottp9;
array pay1tp pay1tp1-pay1tp9;           array pay2tp pay2tp1-pay2tp9;
array pay3tp pay3tp1-pay3tp9;           array pay4tp pay4tp1-pay4tp9;
array pay5tp pay5tp1-pay5tp9;           array odregrhr odregrhr1-odregrhr9;

do i=1 to 9; reghr[i]=0; othr[i]=0; end;

***overtime hours;

do i=1 to 9; /* loop 1. */

if e19200[i] not in (-4,-5) then do;
  if e19200[i]=1 and e24501[i] not in (., -4) then othr[i]=e24501[i];
  if e19200[i] ne 1 and e34403[i] not in (., -4) then othr[i]=e34403[i];
  if e22611[i] not in (., -4) then othr[i]=e22611[i];
end;
if e83100[i] not in (-4,-5) then do;
  if e83100[i]=1 and e88501[i] not in (., -4) then othr[i]=e88501[i];
  if e83100[i] ne 1 and e98403[i] not in (., -4) then othr[i]=e98403[i];
  if e100232[i] not in (., -4) then othr[i]=e100232[i];
end;
if e37901b[i]=1 or e59900[i]=1 then do;
  if e38002[i] not in (., -4) then othr[i]=e38002[i];
  if e38105[i] not in (., -4) then othr[i]=e38105[i];
end;

if othr[i]=0 then newcomp[i]=hrcomp[i];
else if othr[i]<0 then newcomp[i]=-3;
else if othr[i]>0 then do; /*loop 2. */
  new=1;

***get the regular hours;
if e19200[i] not in (-4,-5) then do;
  if e19200[i]=1 and e23901[i] not in (., -4) then reghr[i]=e23901[i];
  if e19200[i] ne 1 and e34402[i] not in (., -4) then reghr[i]=e34402[i];
  if e19200[i] ne 1 and e34428[i] not in (., -4) then reghr[i]=e34428[i];
  if e22610[i] not in (-4,0) then reghr[i]=e22610[i];
end;
if e83100[i] not in (-4,-5) then do;
  if e83100[i]=1 and e88000[i] not in (., -4) then reghr[i]=e88000[i];
  if e83100[i] ne 1 and e98402[i] not in (., -4) then reghr[i]=e98402[i];
  if e83100[i] ne 1 and e98429[i] not in (., -4) then reghr[i]=e98429[i];
  if e100231[i] not in (., -4) then reghr[i]=e100231[i];
end;
if e37901b[i]=1 or e59900[i]=1 then do;
  if e3800ff[i] not in (., -4) then reghr[i]=e3800ff[i];
  if e38103[i] not in (., -4) then reghr[i]=e38103[i];
end;

if reghr[i]<0 then newcomp[i]=-3;
else if reghr[i] ge 0 then do; /*loop 3. */
  ttlhr[i]=reghr[i]+othr[i];

```

Appendix 2: Employment Variable Creation

***decide what hours (reg. hours or total hours or old reg. hours) to multiply by previous calculated hourly
***compensation depending on the time unit for the compensation;

```
pay1hr[i]=reghr[i]; pay2hr[i]=reghr[i]; pay3hr[i]=reghr[i]; pay4hr[i]=reghr[i]; pay5hr[i]=reghr[i];
```

if e19200[i] not in (-4,-5) then do;

```
if e19200[i]=1 and e23901[i] ne -4 then odreghr[i]=e23901[i];
if e19200[i] ne 1 and e34402[i] ne -4 then odreghr[i]=e34402[i];
if e19200[i] ne 1 and e34428[i] ne -4 then odreghr[i]=e34428[i];
if e22610[i] ne -4 then odreghr[i]=e22610[i];
end;
```

if e83100[i] not in (-4,-5) then do;

```
if e83100[i]=1 and e88000[i] ne -4 then odreghr[i]=e88000[i];
if e83100[i] ne 1 and e98402[i] ne -4 then odreghr[i]=e98402[i];
if e83100[i] ne 1 and e98429[i] ne -4 then odreghr[i]=e98429[i];
if e100231[i] ne -4 then odreghr[i]=e100231[i];
end;
```

if (e38103[i] ne -4 or e3800ff[i] ne -4) then do;

```
if othpay1[i] ge 0 and (e38329b[i]=-4 and e384071[i]=-4) then pay1hr[i]=odreghr[i];
/* this is when we didn't fix comp. according to the new hours. */

```

```
if othpay2[i] ge 0 and (e38329b[i]=-4 and e384072[i]=-4) then pay2hr[i]=odreghr[i];
if othpay3[i] ge 0 and (e38329b[i]=-4 and e384073[i]=-4) then pay3hr[i]=odreghr[i];
if othpay4[i] ge 0 and (e38329b[i]=-4 and e384074[i]=-4) then pay4hr[i]=odreghr[i];
if othpay5[i] ge 0 and (e38329b[i]=-4 and e384075[i]=-4) then pay5hr[i]=odreghr[i];
end;
```

if e19200[i] not in (-4,-5) and e20700[i]=1 then do;

```
if e216001[i]=1 or ( e216001[i] in (12,13) and e225301[i] in (1,2)) then pay1hr[i]=ttlhr[i];
if e216002[i]=1 or ( e216002[i] in (12,13) and e225302[i] in (1,2)) then pay2hr[i]=ttlhr[i];
if e216003[i]=1 or ( e216003[i] in (12,13) and e225303[i] in (1,2)) then pay3hr[i]=ttlhr[i];
if e216004[i]=1 or ( e216004[i] in (12,13) and e225304[i] in (1,2)) then pay4hr[i]=ttlhr[i];
if e216005[i]=1 or ( e216005[i] in (12,13) and e225305[i] in (1,2)) then pay5hr[i]=ttlhr[i];
if e22613[i]=1 or ( e22613[i] in (12,13) and e22628c[i] in (1,2)) then pay1hr[i]=tdlhr[i];
if e22614[i]=1 or ( e22614[i] in (12,13) and e22629c[i] in (1,2)) then pay2hr[i]=tdlhr[i];
if e22615[i]=1 or ( e22615[i] in (12,13) and e22630c[i] in (1,2)) then pay3hr[i]=tdlhr[i];
if e22616[i]=1 or ( e22616[i] in (12,13) and e22631c[i] in (1,2)) then pay4hr[i]=tdlhr[i];
if e22617[i]=1 or ( e22617[i] in (12,13) and e22632c[i] in (1,2)) then pay5hr[i]=tdlhr[i];
end;
```

if e83100[i] not in (-4,-5) then do;

```
if e102051[i]=1 or ( e102051[i] in (12,13) and e10214aaa1[i] in (1,2)) then pay1hr[i]=ttlhr[i];
if e102052[i]=1 or ( e102052[i] in (12,13) and e10214aaa2[i] in (1,2)) then pay2hr[i]=ttlhr[i];
if e102053[i]=1 or ( e102053[i] in (12,13) and e10214aaa3[i] in (1,2)) then pay3hr[i]=ttlhr[i];
if e102054[i]=1 or ( e102054[i] in (12,13) and e10214aaa4[i] in (1,2)) then pay4hr[i]=ttlhr[i];
if e102055[i]=1 or ( e102055[i] in (12,13) and e10214aaa5[i] in (1,2)) then pay5hr[i]=ttlhr[i];
if e100234[i]=1 or ( e100234[i] in (12,13) and e100250c[i] in (1,2)) then pay1hr[i]=ttlhr[i];
if e100235[i]=1 or ( e100235[i] in (12,13) and e100251c[i] in (1,2)) then pay2hr[i]=ttlhr[i];
if e100236[i]=1 or ( e100236[i] in (12,13) and e100252c[i] in (1,2)) then pay3hr[i]=ttlhr[i];
if e100237[i]=1 or ( e100237[i] in (12,13) and e100253c[i] in (1,2)) then pay4hr[i]=ttlhr[i];
if e100239[i]=1 or ( e100239[i] in (12,13) and e100254c[i] in (1,2)) then pay5hr[i]=ttlhr[i];
end;
```

if e37901b[i]=1 or e59900[i]=1 then do;

```
if e384071[i]=1 or (e384071[i] in (12,13) and e38416g1[i]=1 and e225301[i] in (1,2))
or (e384071[i] in (12,13) and e38416i1[i] in (1,2)) then pay1hr[i]=ttlhr[i];
if e384072[i]=1 or (e384072[i] in (12,13) and e38416g2[i]=1 and e225302[i] in (1,2))
or (e384072[i] in (12,13) and e38416i2[i] in (1,2)) then pay2hr[i]=ttlhr[i];
if e384073[i]=1 or (e384073[i] in (12,13) and e38416g3[i]=1 and e225303[i] in (1,2))
```

Appendix 2: Employment Variable Creation

```

        or (e384073[i] in (12,13) and e38416i3[i] in (1,2)) then pay3hr[i]=ttlhr[i];
if e384074[i]=1 or (e384074[i] in (12,13) and e38416g4[i]=1 and e225304[i] in (1,2))
        or (e384074[i] in (12,13) and e38416i4[i] in (1,2)) then pay4hr[i]=ttlhr[i];
if e384075[i]=1 or (e384075[i] in (12,13) and e38416g5[i]=1 and e225305[i] in (1,2))
        or (e384075[i] in (12,13) and e38416i5[i] in (1,2)) then pay5hr[i]=ttlhr[i];
if e38329[i]=1 and e20700[i]=1 then do;
    if e212002[i]=1 and (e38329b[i]=1 or ( e38329b[i] in (12,13) and e38329g[i]=1 and e225301[i] in (1,2))
        or (e38329b[i] in (12,13) and e38329i[i] in (1,2)) ) then pay1hr[i]=ttlhr[i];
    if e212003[i]=1 and (e38329b[i]=1 or ( e38329b[i] in (12,13) and e38329g[i]=1 and e225302[i] in (1,2))
        or (e38329b[i] in (12,13) and e38329i[i] in (1,2)) ) then pay2hr[i]=ttlhr[i];
    if e212004[i]=1 and (e38329b[i]=1 or ( e38329b[i] in (12,13) and e38329g[i]=1 and e225303[i] in (1,2))
        or (e38329b[i] in (12,13) and e38329i[i] in (1,2)) ) then pay3hr[i]=ttlhr[i];
    if e212005[i]=1 and (e38329b[i]=1 or ( e38329b[i] in (12,13) and e38329g[i]=1 and e225304[i] in (1,2))
        or (e38329b[i] in (12,13) and e38329i[i] in (1,2)) ) then pay4hr[i]=ttlhr[i];
    if e212006[i]=1 and (e38329b[i]=1 or ( e38329b[i] in (12,13) and e38329g[i]=1 and e225305[i] in (1,2))
        or (e38329b[i] in (12,13) and e38329i[i] in (1,2)) ) then pay5hr[i]=ttlhr[i];
end;
end;

***finally, calculate the corrected total compensation;
hrwgtp[i]=hrwg[i]; if hrwg[i]=-4 then hrwgtp[i]=0;
ottp[i]=ot[i]; if ot[i]=-4 then ottpp[i]=0;
pay1tp[i]=othpay1[i]; if othpay1[i]=-4 then pay1tp[i]=0;
pay2tp[i]=othpay2[i]; if othpay2[i]=-4 then pay2tp[i]=0;
pay3tp[i]=othpay3[i]; if othpay3[i]=-4 then pay3tp[i]=0;
pay4tp[i]=othpay4[i]; if othpay4[i]=-4 then pay4tp[i]=0;
pay5tp[i]=othpay5[i]; if othpay5[i]=-4 then pay5tp[i]=0;

if -4<hrwg[i]<0 or -4<ot[i]<0 or -4<othpay1[i]<0 or -4<othpay2[i]<0 or -4<othpay3[i]<0
    or -4<othpay4[i]<0 or -4<othpay5[i]<0 then newcomp[i]=-3;
else newcomp[i]=(hrwgtp[i]*regrh[i]+ottp[i]*othr[i]+pay1tp[i]*pay1hr[i]+pay2tp[i]*pay2hr[i]+pay3tp[i]*pay3hr[i]
    +pay4tp[i]*pay4hr[i]+pay5tp[i]*pay5hr[i])/ttlhr[i];

end; /* end of loop 3. */

end; /* end of loop 2. */

end; /* end of loop 1. */

/***************** job lasts 13 weeks or less *****/
jleng01=-4;      jleng02=-4;      jleng03=-4;      jleng04=-4;      jleng05=-4;
jleng06=-4;      jleng07=-4;      jleng08=-4;      jleng09=-4;
array jleng jleng01 jleng02 jleng03 jleng04 jleng05 jleng06 jleng07 jleng08 jleng09 jleng10;

do i=1 to 9;
    if e37701[i]=0 then jleng[i]=0;
    if e58201[i]=0 then jleng[i]=0;
    if e37901b[i]>-1 or e59900[i]>-1 or e58401[i]>-1 then do;
        if e37901b[i]=1 or e59900[i]=1 or e58401[i]=0 then jleng[i]=1;
        else jleng[i]=0;
    end;
end;

/*****************missing values*****************/
array jleng jleng01 jleng02 jleng03 jleng04 jleng05 jleng06 jleng07 jleng08 jleng09;
array hrwg hrwg01 hrwg02 hrwg03 hrwg04 hrwg05 hrwg06 hrwg07 hrwg08 hrwg09 ;
array newcomp newcomp1-newcomp9;

```

```
do i=1 to 9;
  if jleng[i]>-4 and hrwg[i]=-4 then hrwg[i]=-3;
  if jleng[i]>-4 and newcomp[i]=-4 then newcomp[i]=-3;
end;

array hrwgr hrwgr01 hrwgr02 hrwgr03 hrwgr04 hrwgr05 hrwgr06 hrwgr07 hrwgr08 hrwgr09;
hrwgr01=-4;      hrwgr02=-4;      hrwgr03=-4;      hrwgr04=-4;      hrwgr05=-4;
hrwgr06=-4;      hrwgr07=-4;      hrwgr08=-4;      hrwgr09=-4;

do i=1 to 9;
  hrwgr[i]=round(hrwg[i],1);
  if e200a=-5 then hrwgr[i]=-5;
  if e200a=-5 then jleng[i]=-5;
end;

array hrcomr hrcomr01 hrcomr02 hrcomr03 hrcomr04 hrcomr05 hrcomr06 hrcomr07 hrcomr08 hrcomr09;
hrcomr01=-4;      hrcomr02=-4;      hrcomr03=-4;      hrcomr04=-4;      hrcomr05=-4;
hrcomr06=-4;      hrcomr07=-4;      hrcomr08=-4;      hrcomr09=-4;

do i=1 to 9;
  hrcomr[i]=round(newcomp[i],1);
  if e200a=-5 then hrcomr[i]=-5;
end;

/*hand edit*/ if id=1280802 then do; hrcomr02=-3; hrwgr02=-3; end;

endsas;
```

